ALBERT-LUDWIGS-UNIVERSITÄT FREIBURG

Algorithms for Radio Networks

Routing, Flooding, DSR

University of Freiburg
Technical Faculty
Computer Networks and Telematics
Christian Schindelhauer

CoNe Freiburg

IIF INSTITUT FÜR INFORMATIK FREIBURG

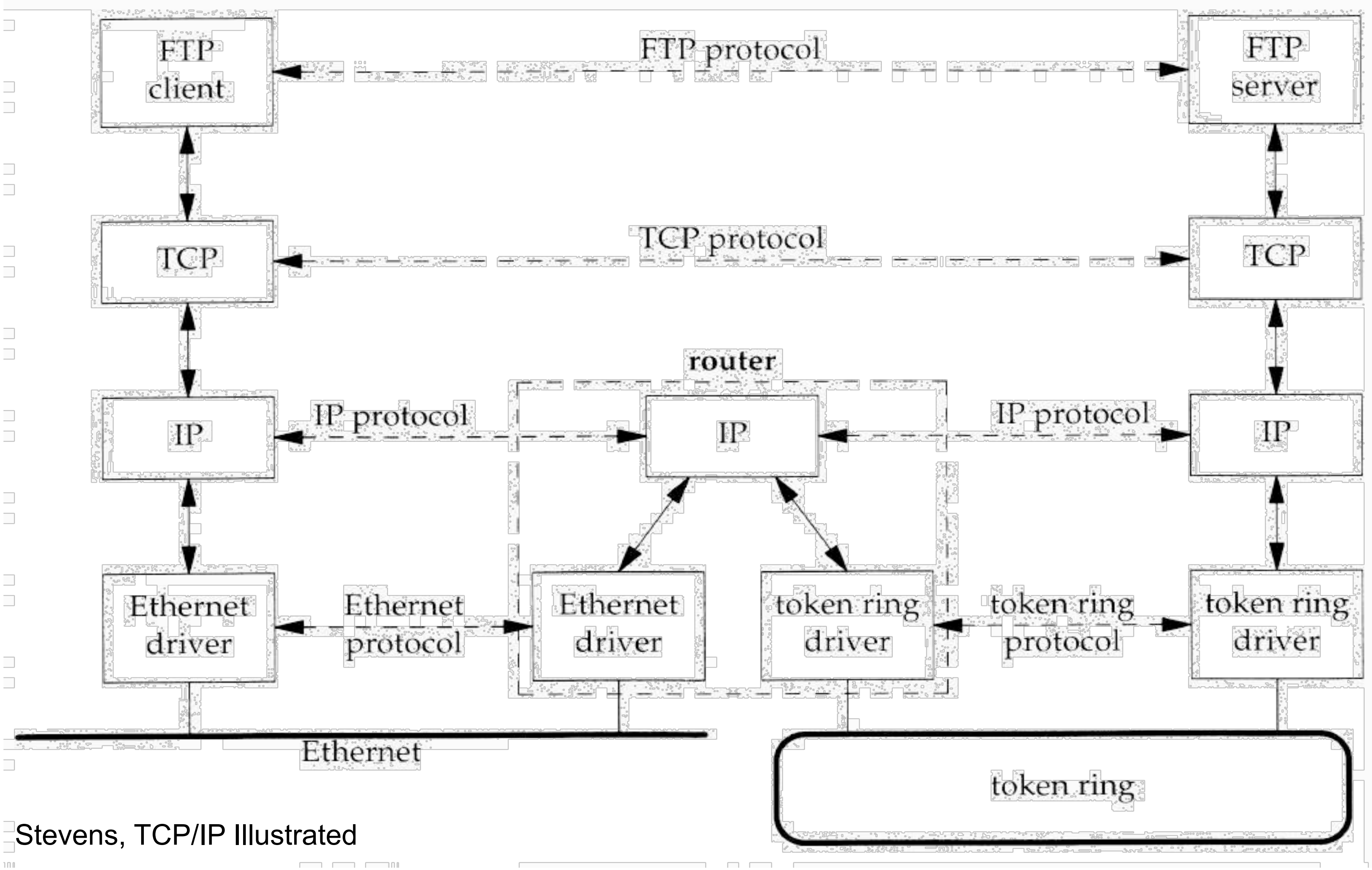
Protocols of the Internet

Application	Telnet, FTP, HTTP, SMTP (E-Mail), ...
Transport	TCP (Transmission Control Protocol) UDP (User Datagram Protocol)
Network	IP **(Internet Protocol)** + ICMP **(Internet Control Message Protocol)** + IGMP **(Internet Group Management Protocol)**
Host-to-Network	LAN **(e.g. Ethernet, Token Ring etc.)**

TCP/IP Layers

- **1. Host-to-Network**
  - Not specified, depends on the local network,k e.g. Ethernet, WLAN 802.11, PPP, DSL
- **2. Routing Layer/Network Layer (IP - Internet Protocol)**
  - Defined packet format and protocol
  - Routing
  - Forwarding
- **3. Transport Layer**
  - TCP (Transmission Control Protocol)
    - Reliable, connection-oriented transmission
    - Fragmentation, Flow Control, Multiplexing
  - UDP (User Datagram Protocol)
    - hands packets over to IP
    - unreliable, no flow control
- **4. Application Layer**
  - Services such as TELNET, FTP, SMTP, HTTP, NNTP (for DNS), ...

Example: Routing between LANs



Stevens, TCP/IP Illustrated

Routing Tables and Packet Forwarding

- **IP Routing Table**
  - contains for each destination the address of the next gateway
  - destination: host computer or sub-network
  - default gateway
- **Packet Forwarding**
  - IP packet (datagram) contains start IP address and destination IP address
    - if destination = my address then hand over to higher layer
    - if destination in routing table then forward packet to corresponding gateway
    - if destination IP subnet in routing table then forward packet to corresponding gateway
    - otherwise, use the default gateway

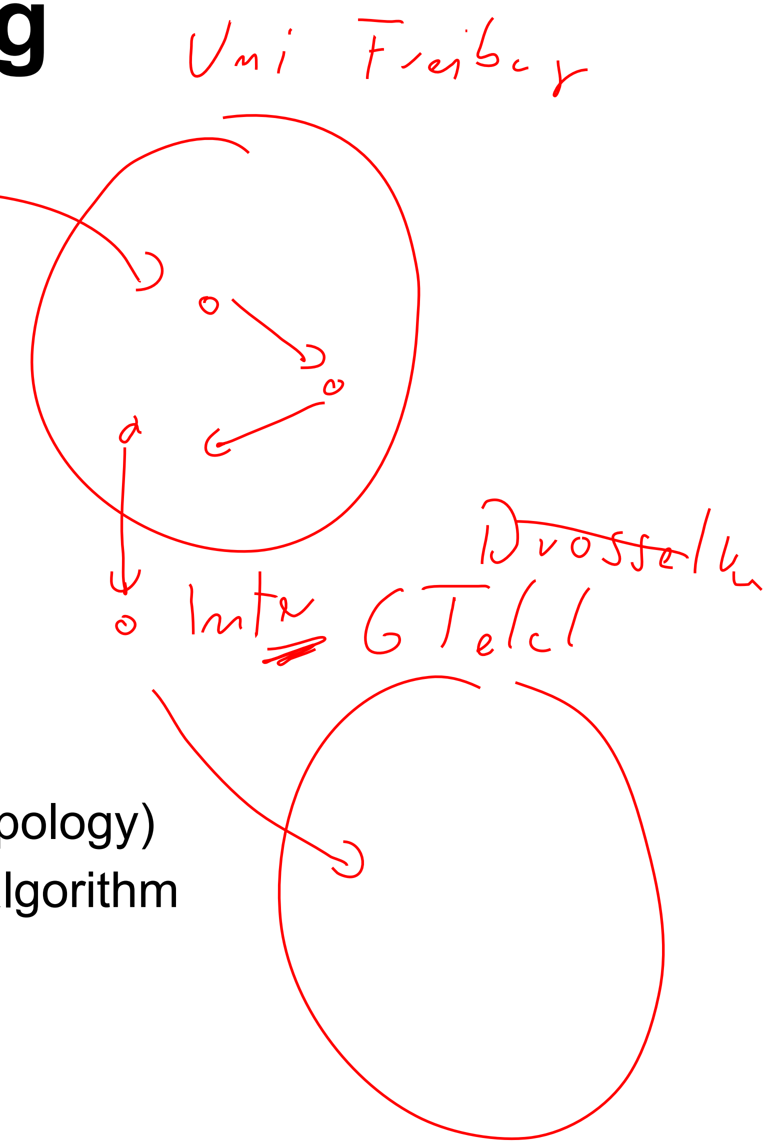
IP Packet Forwarding

- **IP -Packet (datagram) contains...**
  - TTL (Time-to-Live): Hop count limit
  - Start IP Address
  - Destination IP Address
- **Packet Handling**
  - Reduce TTL (Time to Live) by 1
  - If TTL ≠ 0 then forward packet according to routing table
  - If TTL = 0 or forwarding error (buffer full etc.):
    - delete packet
    - if packet is not an ICMP Packet then
      - send ICMP Packet with
        - start = current IP Address
        - destination = original start IP Address

Static and Dynamic Routing

- **Static Routing**
  - Routing table created manually
  - used in small LANs
- **Dynamic Routing**
  - Routing table created by Routing Algorithm
  - Centralized, e.g. Link State
    - Router knows the complete network topology
  - Decentralized, e.g. Distance Vector
    - Router knows gateways in its local neighborhood

Intra-AS Routing

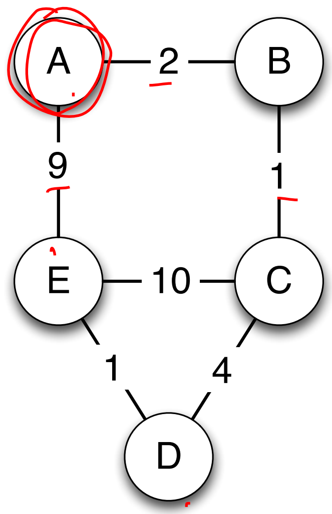


- **Routing Information Protocol (RIP)**
  - Distance Vector Algorithmus
  - Metric = hop count
  - exchange of distance vectors (by UDP)
- **Interior Gateway Routing Protocol (IGRP)**
  - successor of RIP
  - different routing metrics (delay, bandwidth)
- **Open Shortest Path First (OSPF)**
  - Link State Routing (every router knows the topology)
  - Route calculation by Dijkstra's shortest path algorithm

Distance Vector Routing Protocol

shortest path

- **Distance Table data structure**
  - Each node has a
    - Line for each possible destination
    - Column for any direct neighbors
- **Distributed algorithm**
  - each node communicates only with its neighbors
- **Asynchronous operation**
  - Nodes do not need to exchange information in each round
- **Self-terminating**
  - exchange unless no update is available



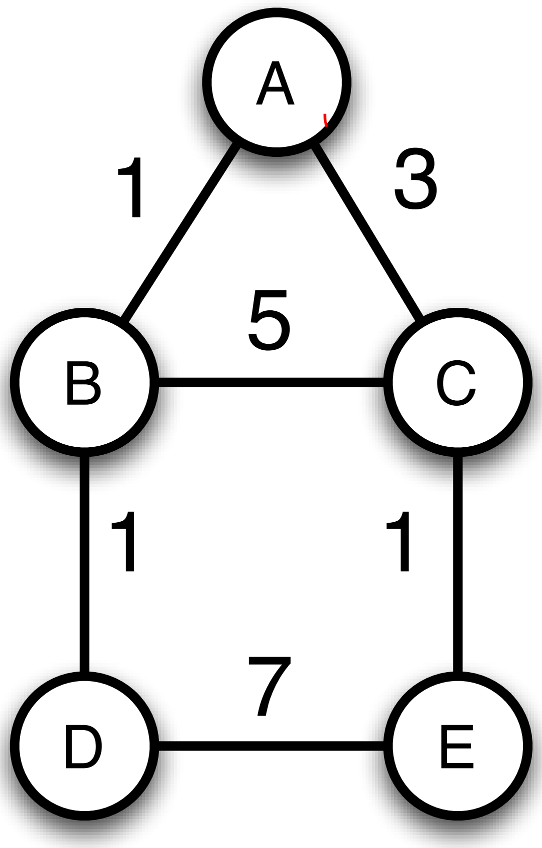
Distance Table for A

from A		via B	via E	Routing Table entry
to	B	**2**	15	B
	C	**3**	14	B
	D	**7**	10	B
	E	**8**	9	E

Distance Table for C

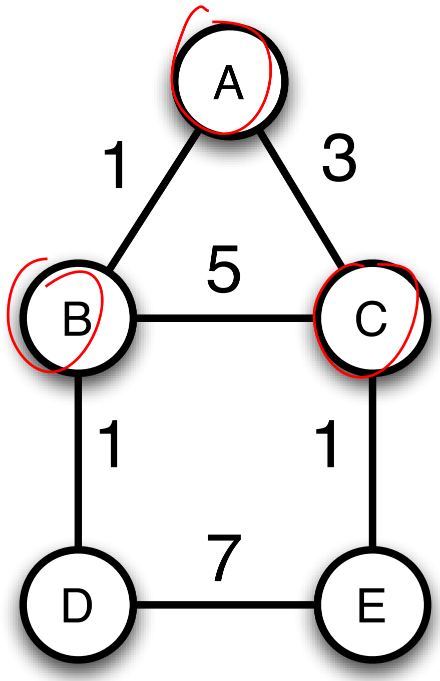
from C		via B	via D	via E	Routing Table entry
to	A	**3**	11	18	B
	B	**1**	9	21	B
	D	6	**4**	11	D
	E	7	**5**	10	D

Distance Vector Routing Example



from A to	via B	via C	entry
B	**1**	8	**B**
C	6	**3**	**C**
D	**2**	9	**B**
E	7	**4**	**C**

Distance Vector Routing



from A to	via		entry
	B	C	
B	1	∞ -	B
C	∞ -	3	C
D	-	-	-
E	-	-	-

from B to	via			entry
	A	C	D	
A	1	-	-	A
C	-	3	-	C
D	-	-	1	C
E	-	-	~~8~~	D

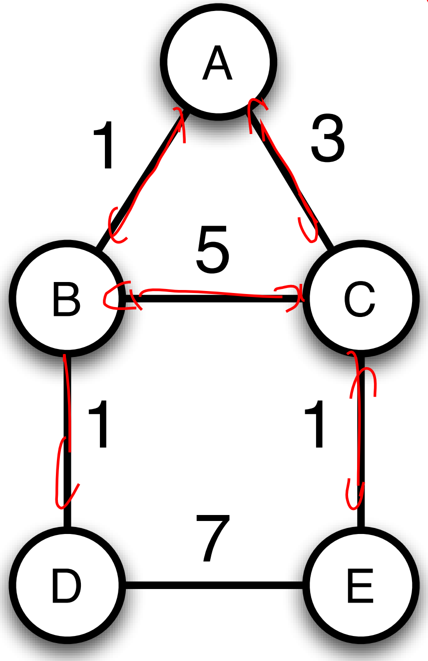
from C to	via			entry
	A	B	E	
A	3	-	-	A
B	-	5	-	B
D	-	-	8	E
E	-	-	1	E

Distance Vector Routing

from B to	via A	via C	via D	Entry
A	1	-	-	A
C	-	5	-	C
D	-	-	1	D
E	-	-	8	D

from C to	via A	via B	via E	Entry
A	3	-	-	A
B	-	5	-	B
D	-	-	8	E
E	-	-	1	E

from B to	via A	via C	via D	Entry
A	1	8	-	A
C	-	5	-	C
D	-	13	1	D
E	-	6	8	C

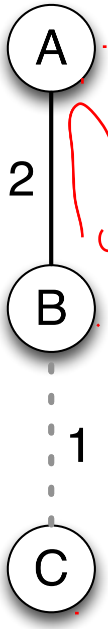


from C to	via A	via B	via E	Entry
A	3	6	-	A
B	-	5	-	B
D	-	6	8	B
E	-	13	1	E

“Count to Infinity” - Problem

- **Good news travels fast**
  - A new connection is quickly at hand
- **Bad news travels slowly**
  - Connection fails
  - Neighbors increase their distance mutally
  - "Count to Infinity" Problem

“Count to Infinity” - Problem



from A	via B	Routing Table entry
to B	2	B
to C	3	B

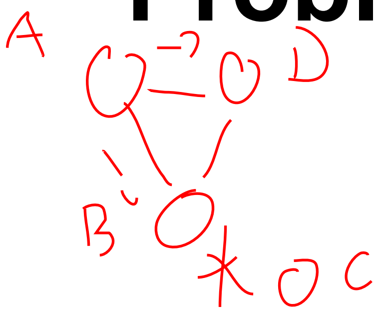
from B	via A	via C	Routing Table entry
to A	2	-	A
to C	5	-	A

from A	via B	Routing Table entry
to B	2	B
to C	7	B

from B	via A	via C	Routing Table entry
to A	2	-	A
to C	5	-	A

from A	via B	Routing Table entry
to B	2	B
to C	7	B

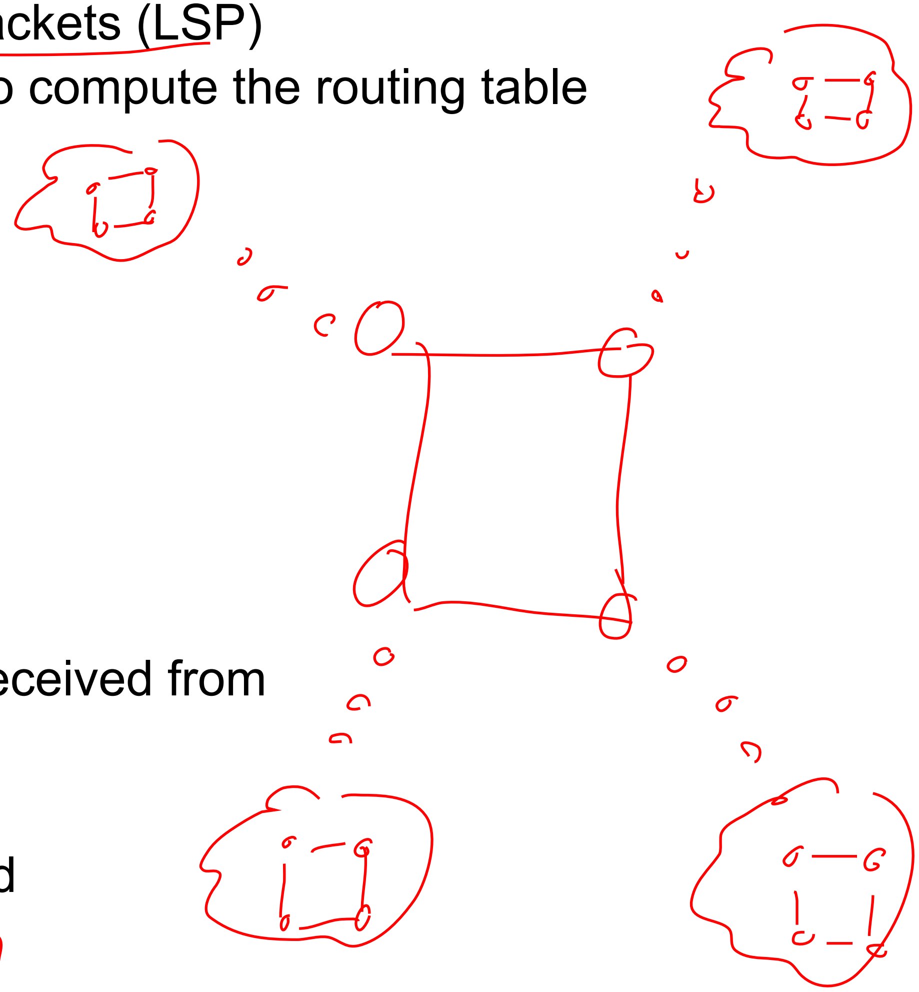
from B	via A	via C	Routing Table entry
to A	2	-	A
to C	9	-	A

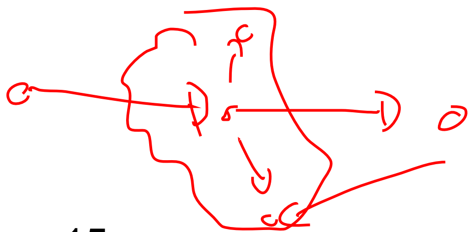


for B	via A	via C
to A	2	4
C	5	1

Link-State Protocol

- **Link state routers**
  - exchange information using Link State Packets (LSP)
  - each node uses shortest path algorithm to compute the routing table
- **LSP contains**
  - ID of the node generating the packet
  - Cost of this node to any direct neighbors
  - Sequence-no. (SEQNO)
  - TTL field for that field (time to live)
- **Reliable flooding (Reliable Flooding)**
  - current LSP of each node are stored
  - Forward of LSP to all neighbors
    - except to be node where it has been received from
  - Periodically creation of new LSPs
    - with increasing SEQNO
  - Decrement TTL when LSPs are forwarded





Characteristics of routing in mobile ad hoc networks

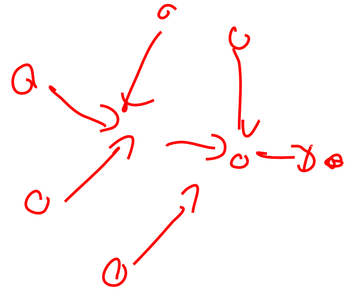
- **Movement of participants**
  - Reconnecting and loss of connection is more common than in other wireless networks
  - Especially at high speed
- **Other performance criteria**
  - Route stability in the face of mobility
  - energy consumption

Unicast Routing

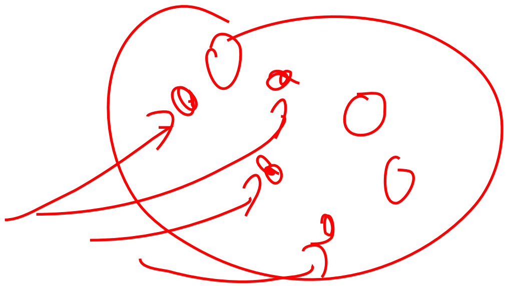
Point to Point

- **Variety of protocols**
  - Adaptations and new developments
- **No protocol dominates the other in all situations**
  - Solution: Adaptive protocols?

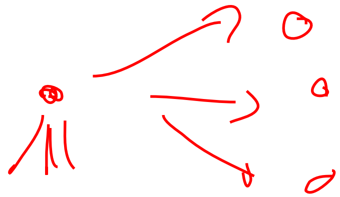
Convergecast



Multicast



Broadcast



all

Routing in MANETs

- **Routing**
  - Determination of message paths
  - Transport of data
- **Protocol types**
  - proactive
    - Routing tables with updates
  - reactive
    - repair~~m~~ of message paths only when necessary
  - hybrid
    - combination of proactive and reactive

Routing Protocols

- **Proactive**
  - Routes are **demand independent**
  - Standard Link-State und Distance-Vector Protocols
    - Destination Sequenced Distance Vector (**DSDV**)
    - Optimized Link State Routing (**OLSR**)
- **Reactive**
  - Route are determined when needed
    - Dynamic Source Routing (**DSR**)
    - Ad hoc On-demand Distance Vector (**AODV**)
    - Dynamic MANET On-demand Routing Protocol DYMO
    - Temporally Ordered Routing Algorithm (**TORA**)
- **Hybrid**
  - combination of reactive und proactive
    - Zone Routing Protocol (**ZRP**)
    - Greedy Perimeter Stateless Routing (**GPSR**)

many more

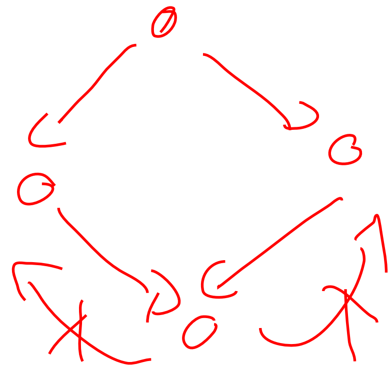
Source Route

Trade-Off

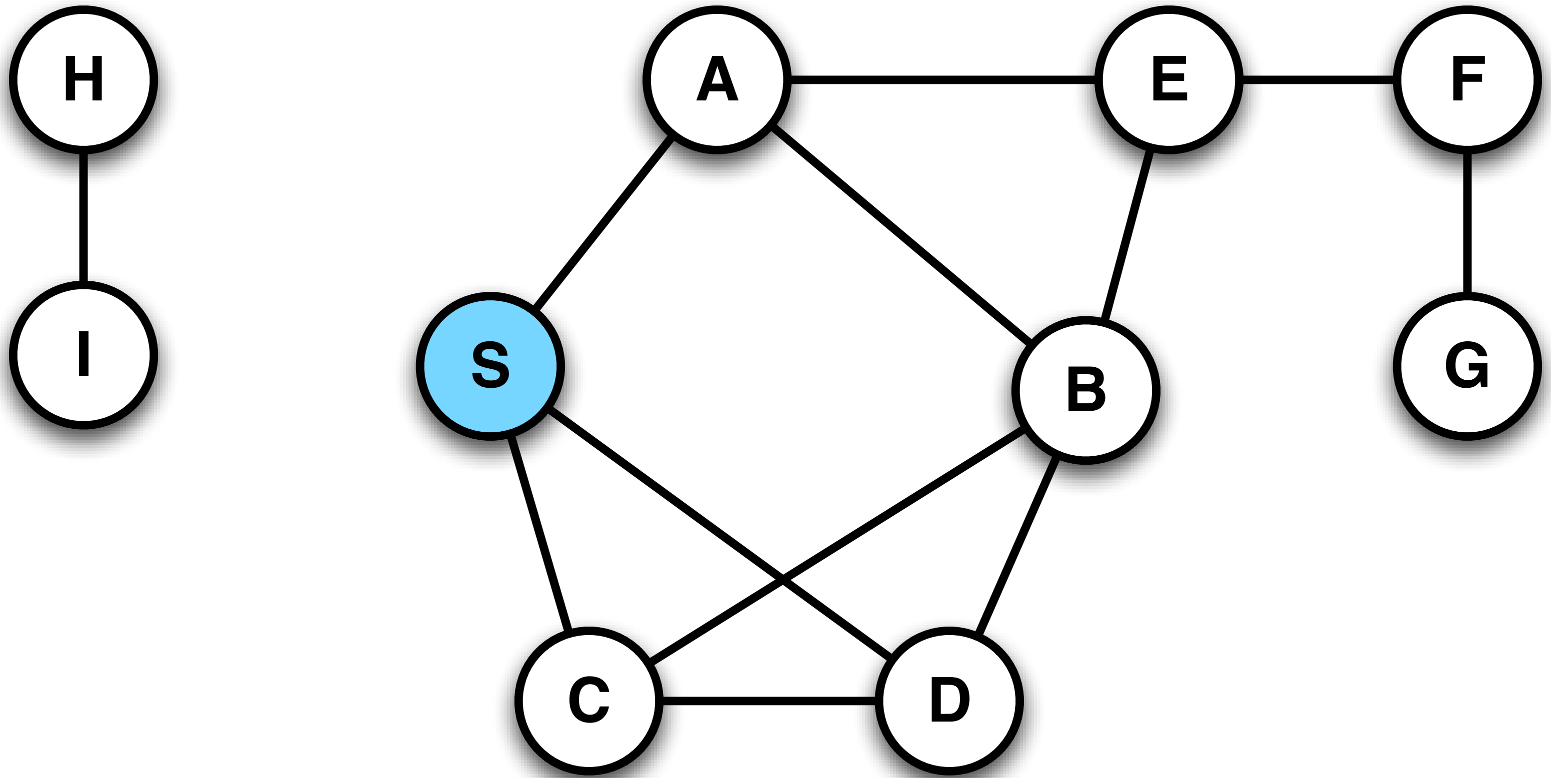
- **Laten~~z~~cy because of route discovery**
  - Proactive protocols are faster
  - Reactive protocols need to find routes
- **Overhead of Route discovery and maintenance**
  - Reactive protocols have smaller overhead (number of messages)
  - Proactive protocols may have larger complexity
- **Traffic-Pattern and mobility**
  - decides which type of protocol is more efficient

Flooding

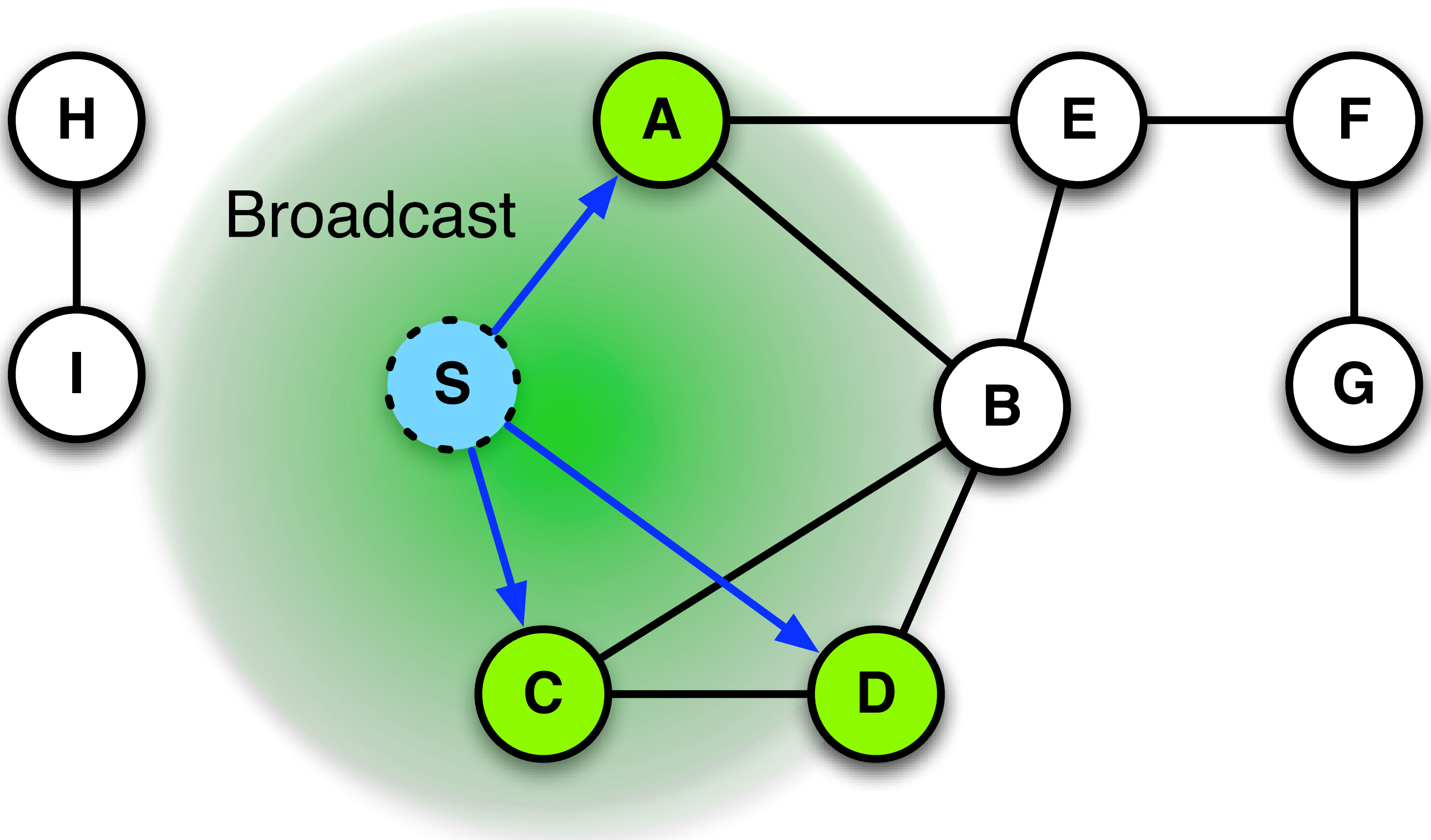
- **Algorithm**
  - Sender S broadcasts data packet to all neighbors
  - Each node receiving a new packet
    - broadcasts this packet
    - if it is not the receiver
- **Sequence numbers**
  - identifies messages to prevent duplicates
- **Packet always reaches the target**
  - if possible



Flooding Example

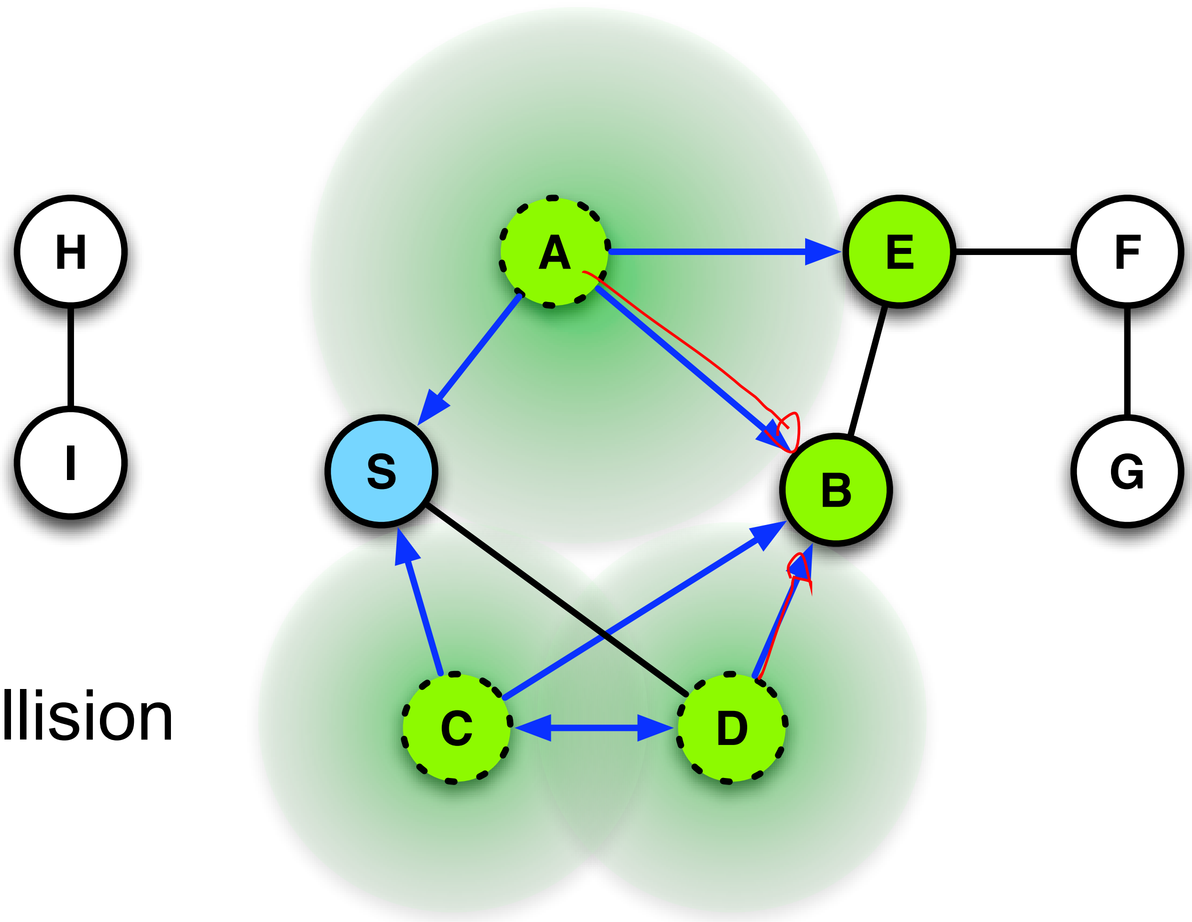


Flooding for Data Delivery



Packet for Receiver F

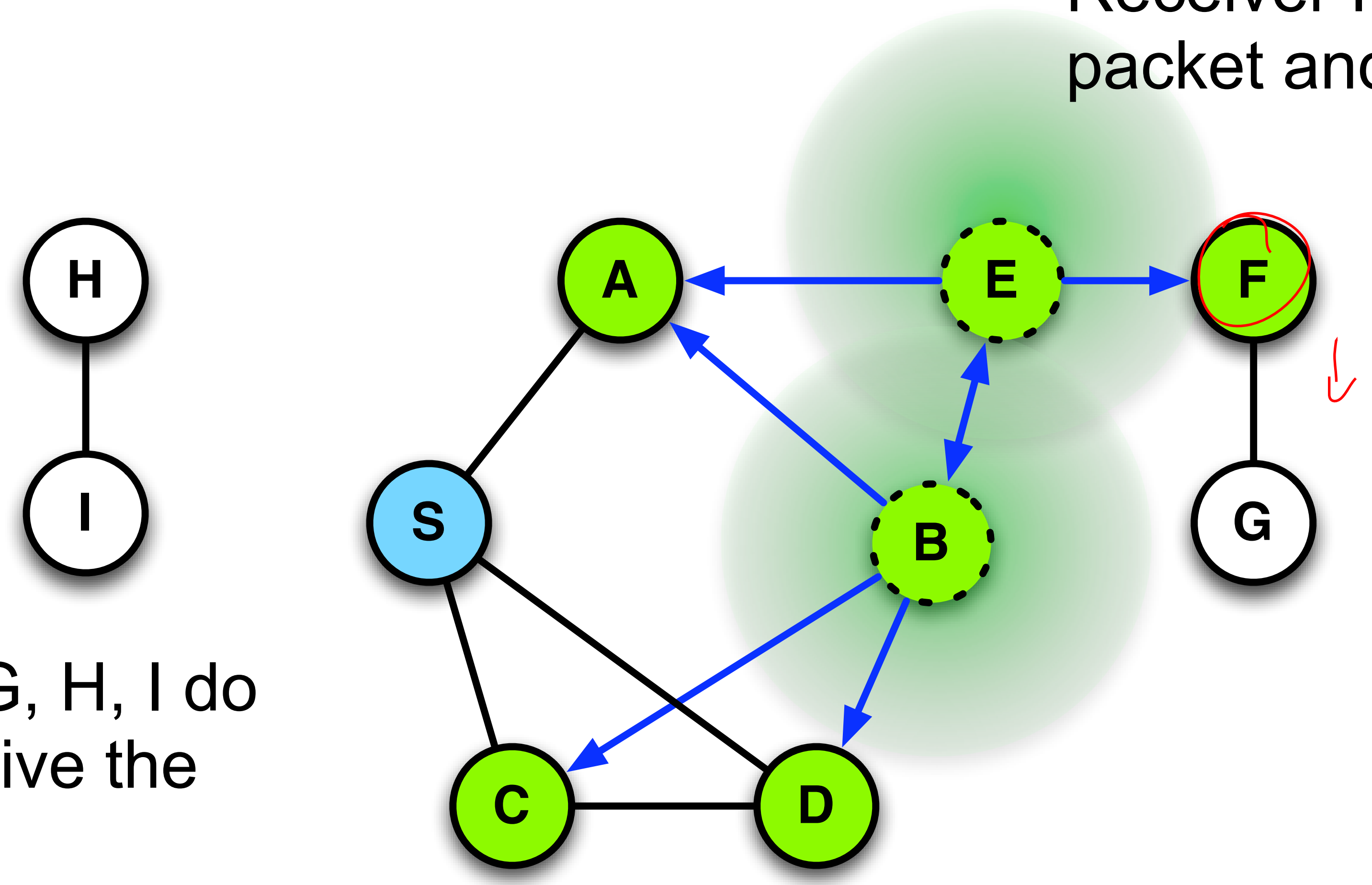
Flooding for Data Delivery



Possible collision at B

Flooding for Data Delivery

Receiver F gets packet and stops



Nodes G, H, I do not receive the packet

Flooding

- **Advantage**
  - simple and robust
  - the best approach for short packet lengths, small number of participants in highly mobile networks with light traffic
- **Disadvantage**
  - High overhead
  - Broadcasting is unreliable
    - lack of acknowledgements
    - hidden, exposed terminals lead to data loss or delay

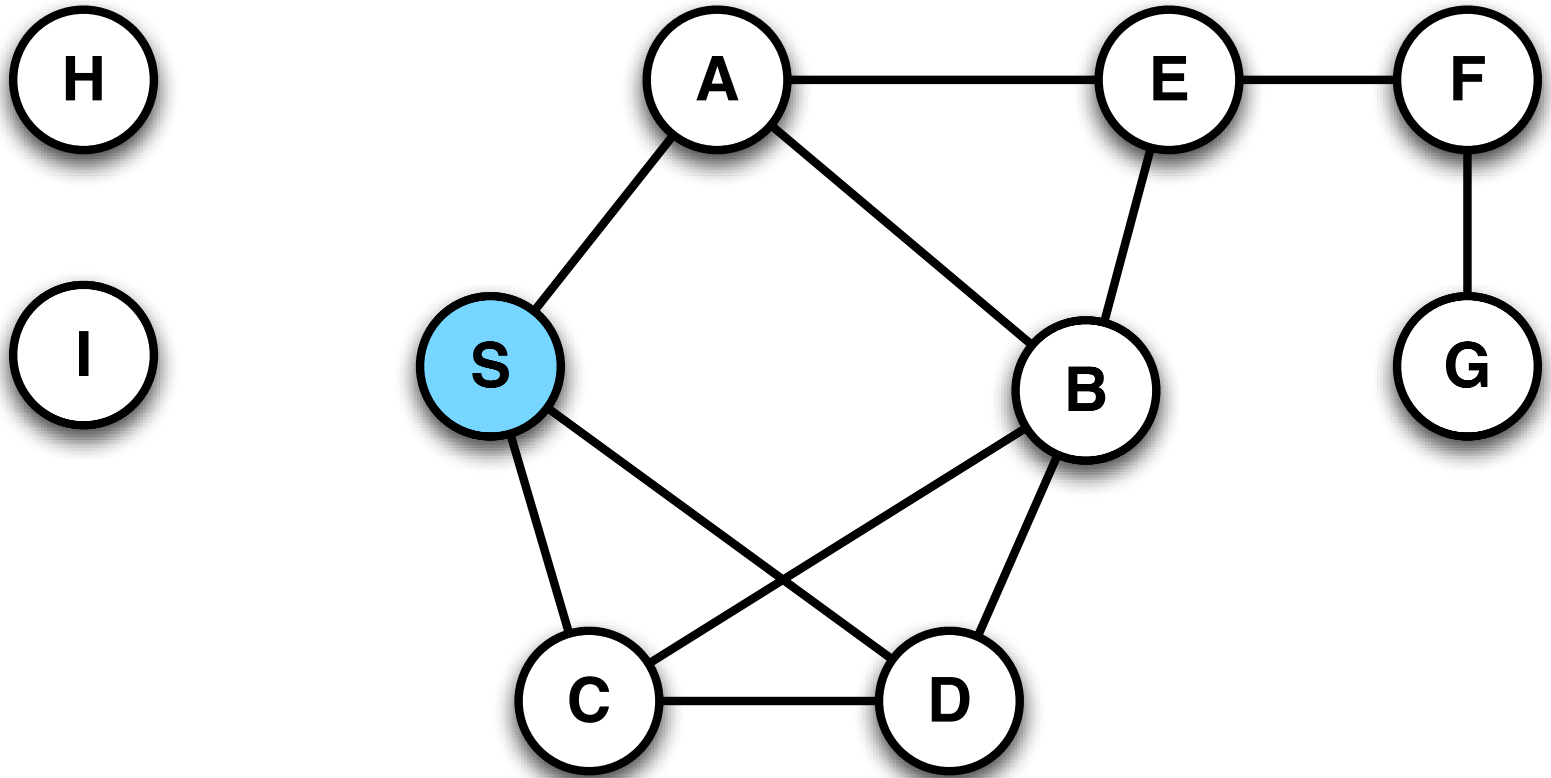
Flooding

- **Produces too many unnecessary (long) data packets**
  - in the worst case, each participant sends each packet
  - many long transmissions collisions lead to long waiting times in the medium access
- **Better approach:**
  - Use of control packets for route determination
  - Flooding of control packet leads to DSR

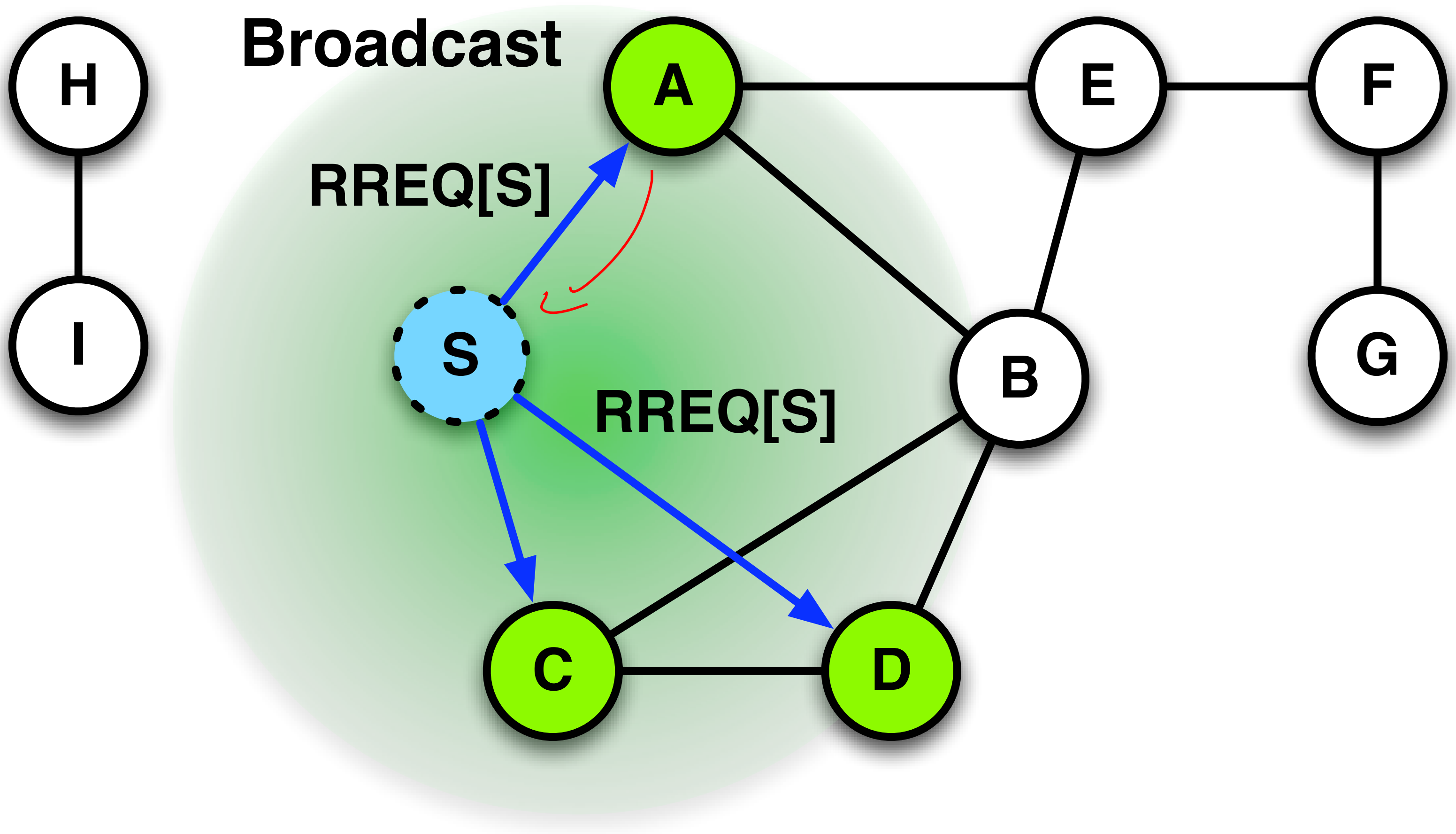
Dynamic Source Routing (DSR)

- **Johnson, Maltz**
  - *Dynamic Source Routing in Ad Hoc Wireless Networks*, Mobile Computing, 1996
- **Algorithm**
  - Sender initiates route discovery by flooding of **Route-Request (RREQ)**-packets
    - Each forwarding node appends his ID to the RREQ-packet
  - The receiver generates the routing information from the RREQ packet by producing a **Route-Reply (RREP)-**packet
    - using the route information of the packet is sent back to the sender
  - Transmitter sends **data packet** along with route information to the receiver

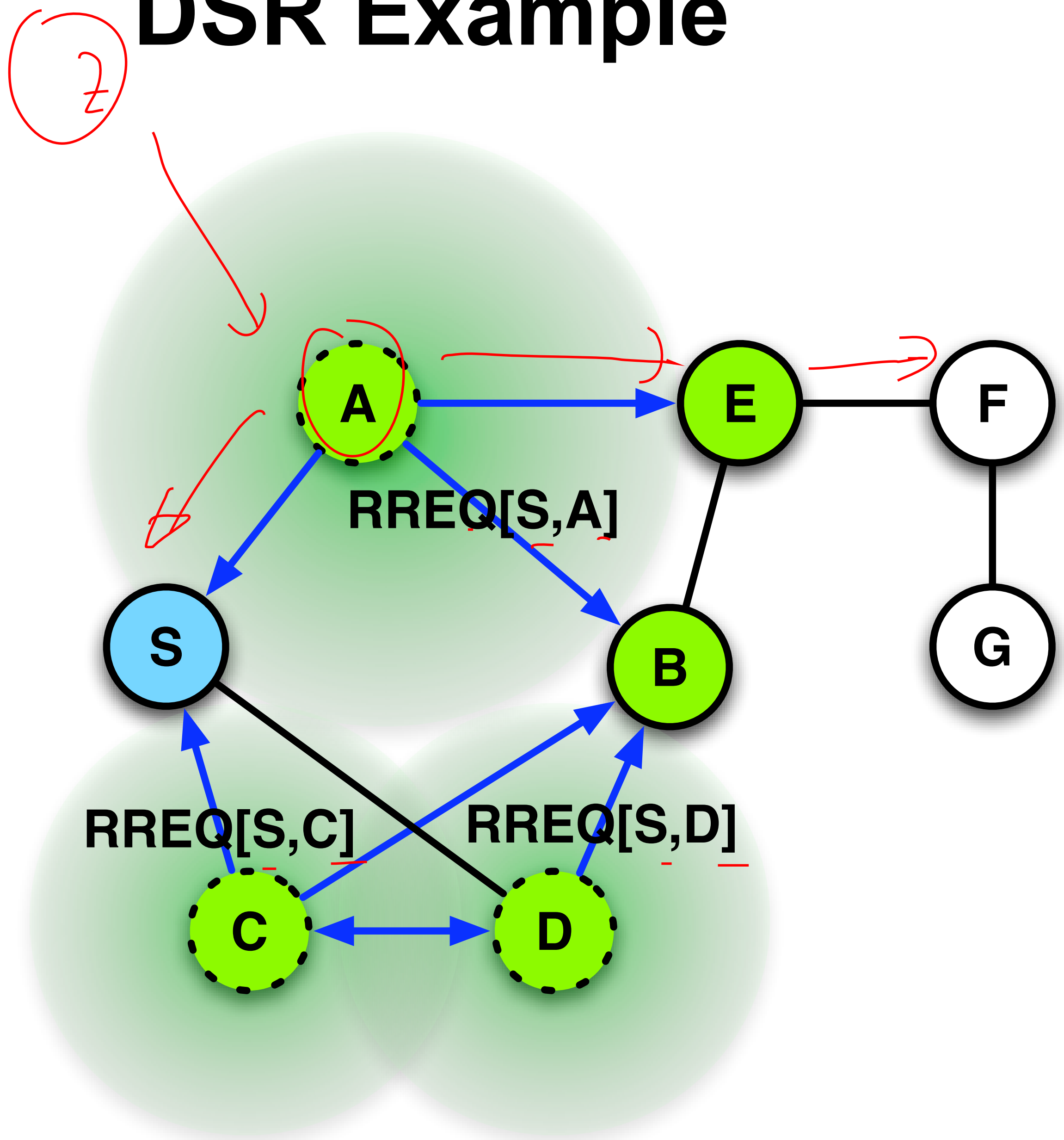
DSR Example



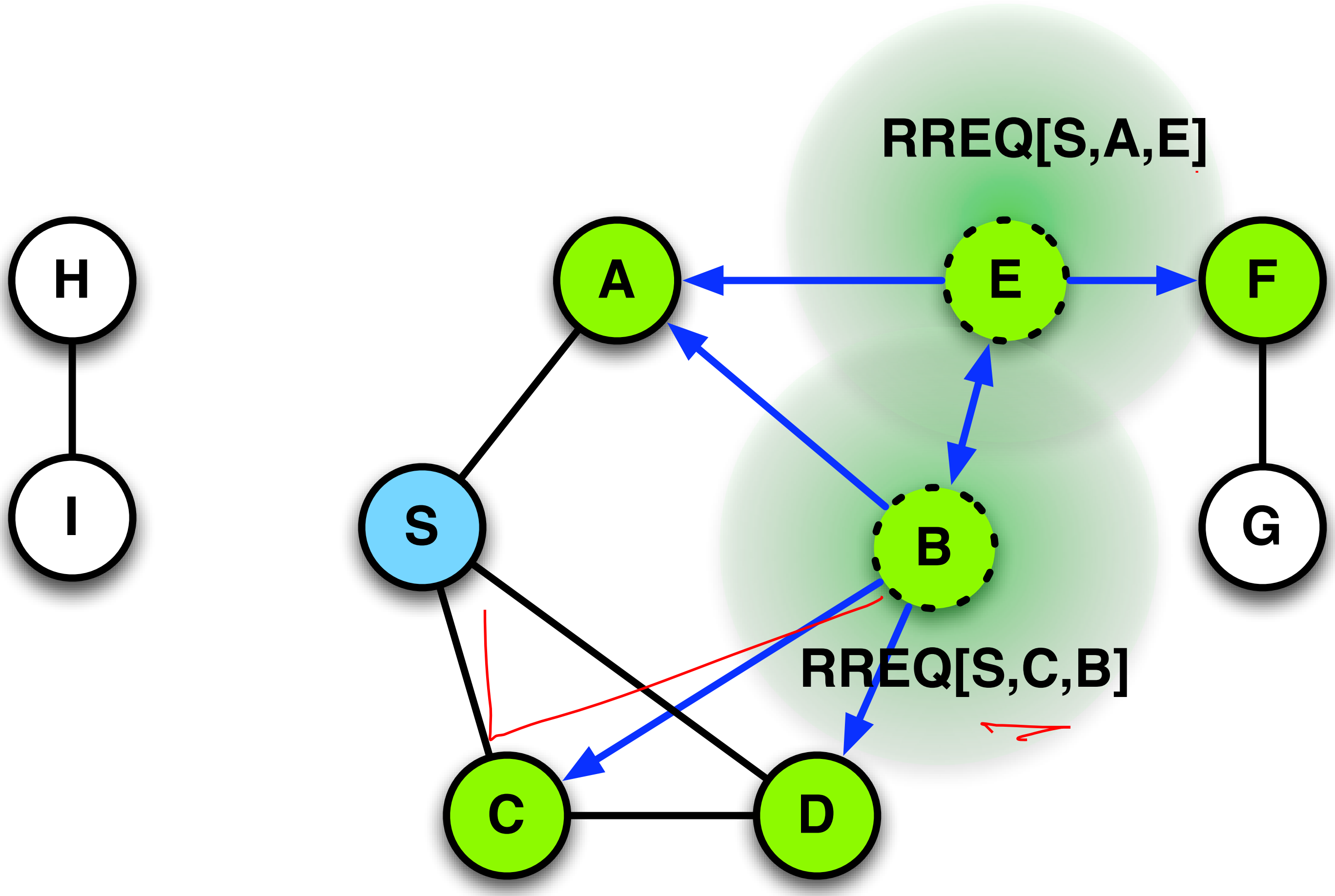
DSR Example



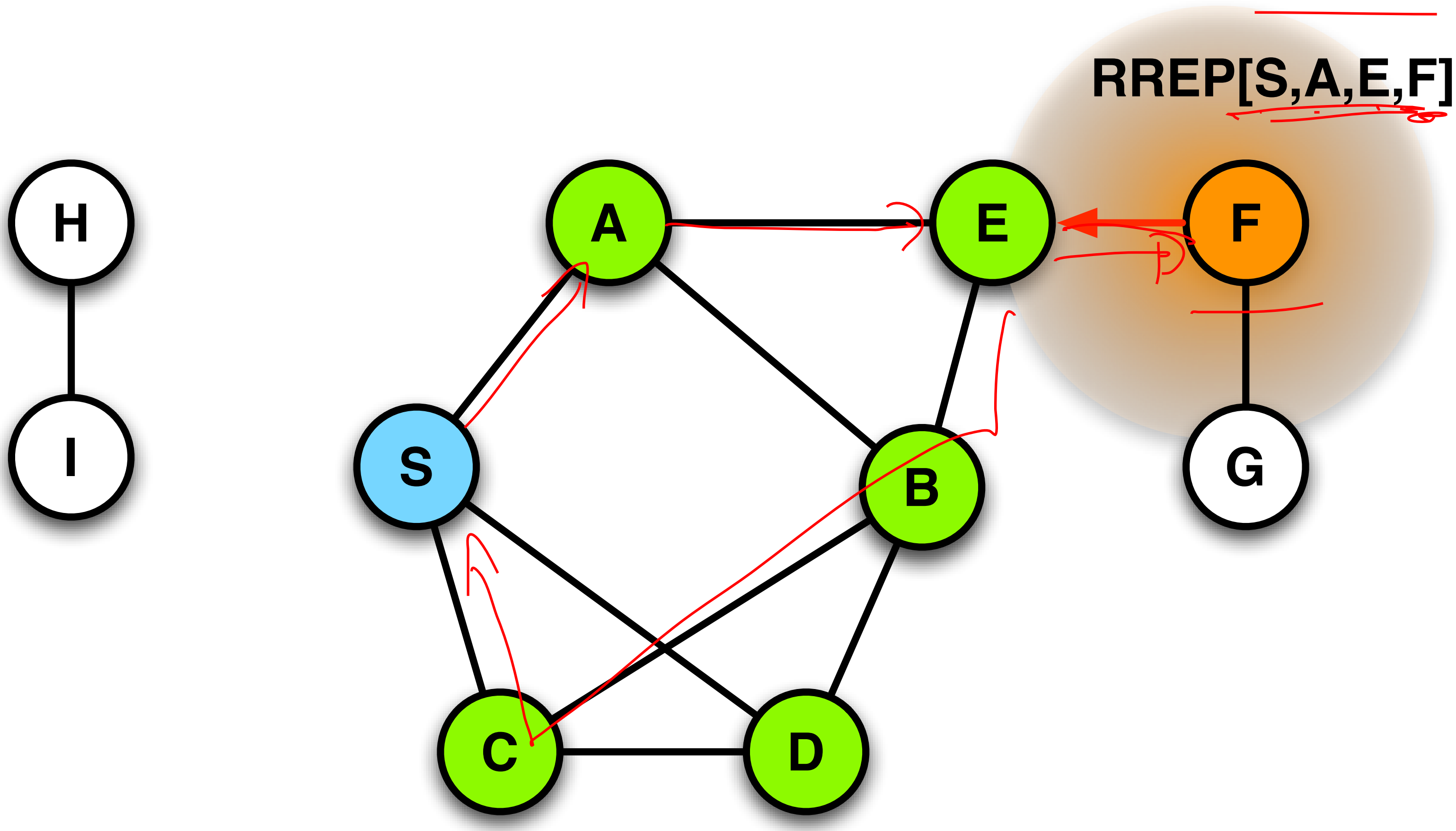
DSR Example



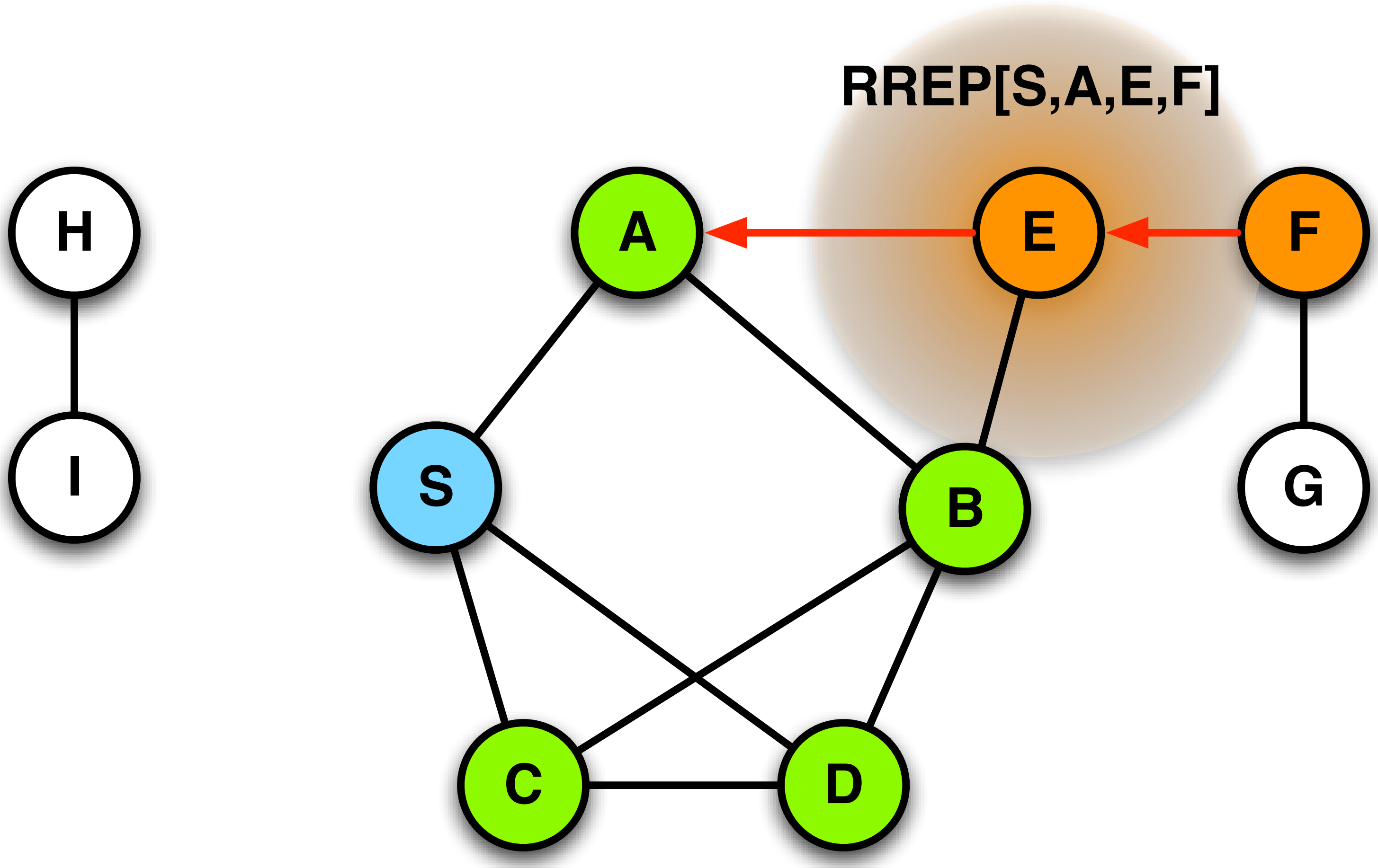
DSR Example



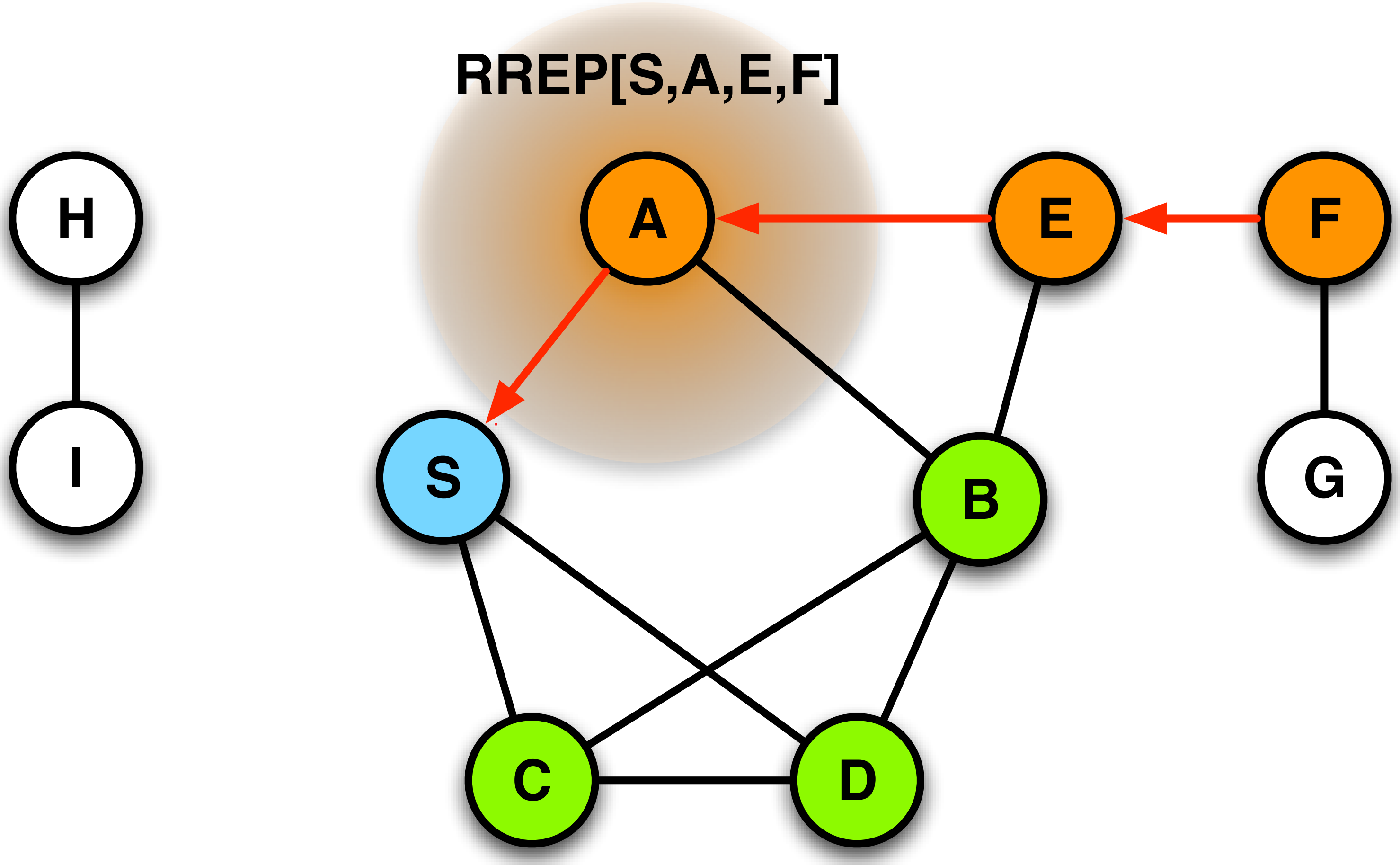
DSR Example



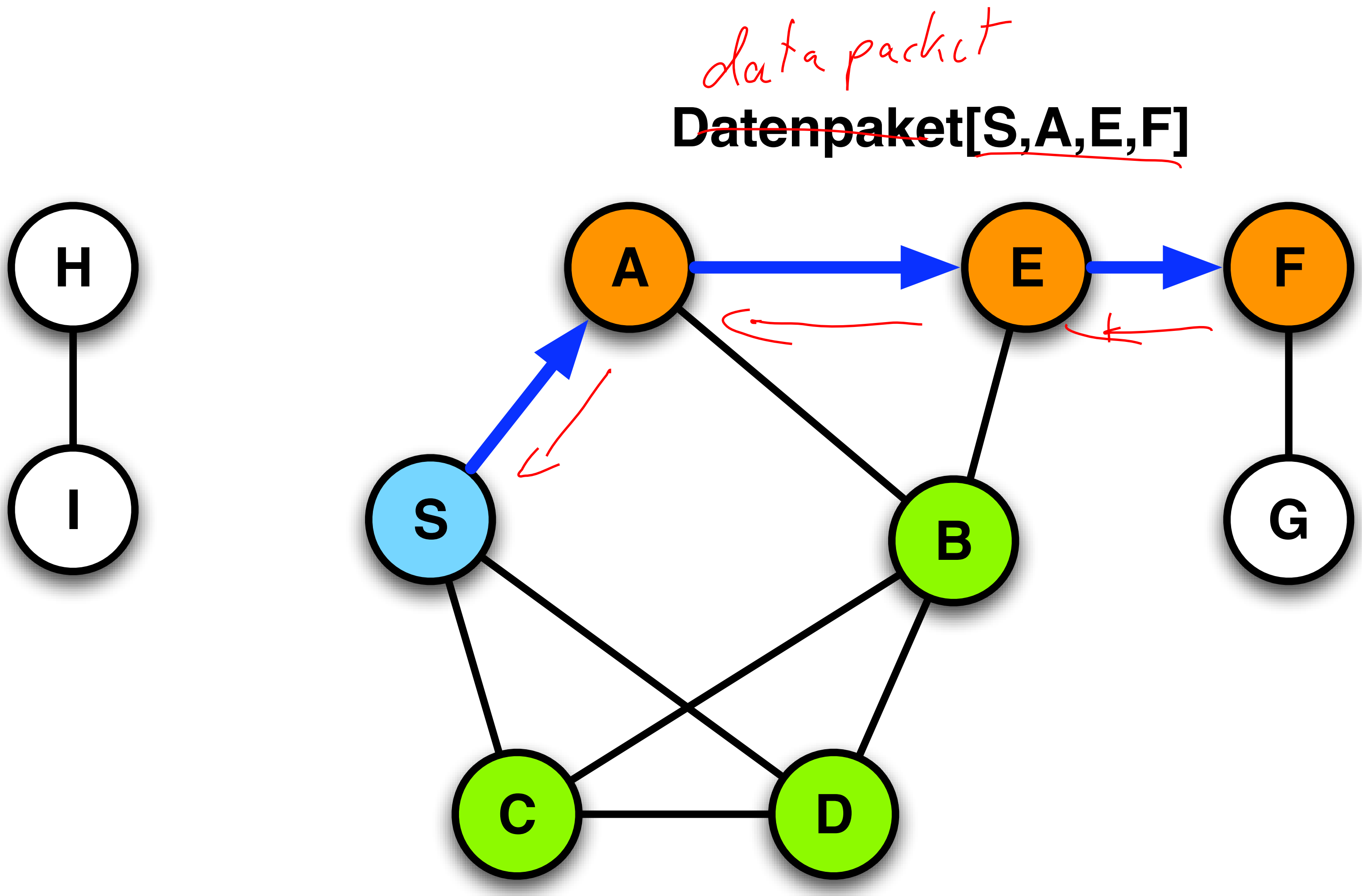
DSR Example



DSR Example



DSR Example



Requirements

- **Route Reply**
  - requires bidirectional connections
  - unidirectional links
    - must be tested for symmetry
    - or Route-Reply must trigger its own route-request
- **Data packet has all the routing information in the header**
  - hence: Source-Routing
- **Route determination**
  - if no valid route is known

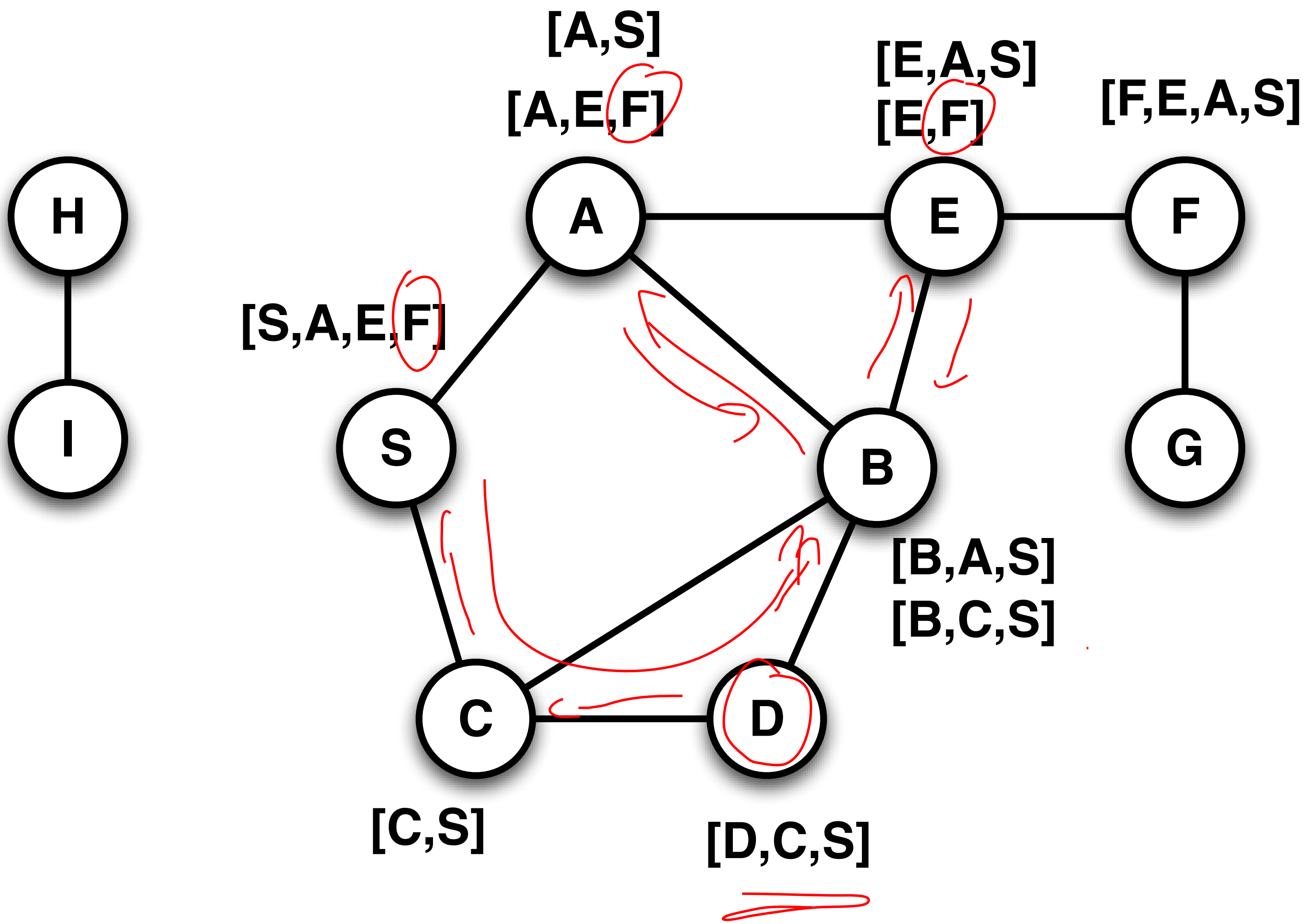
DSR Extensions and Modifications

- **Intermediate nodes can cache information RREP**
  - Problem: stale information
- **Listening to control messages**
  - can help to identify the topology
- **Random delays for answers**
  - To prevent many RREP-packets (Reply-Storm)
  - if many nodes know the answer (not for media access)
- **Repair**
  - If an error is detected then usually: route recalculation
  - Instead: a local change of the source route
- **Cache Management**
  - Mechanisms for the deletion of outdated cache information

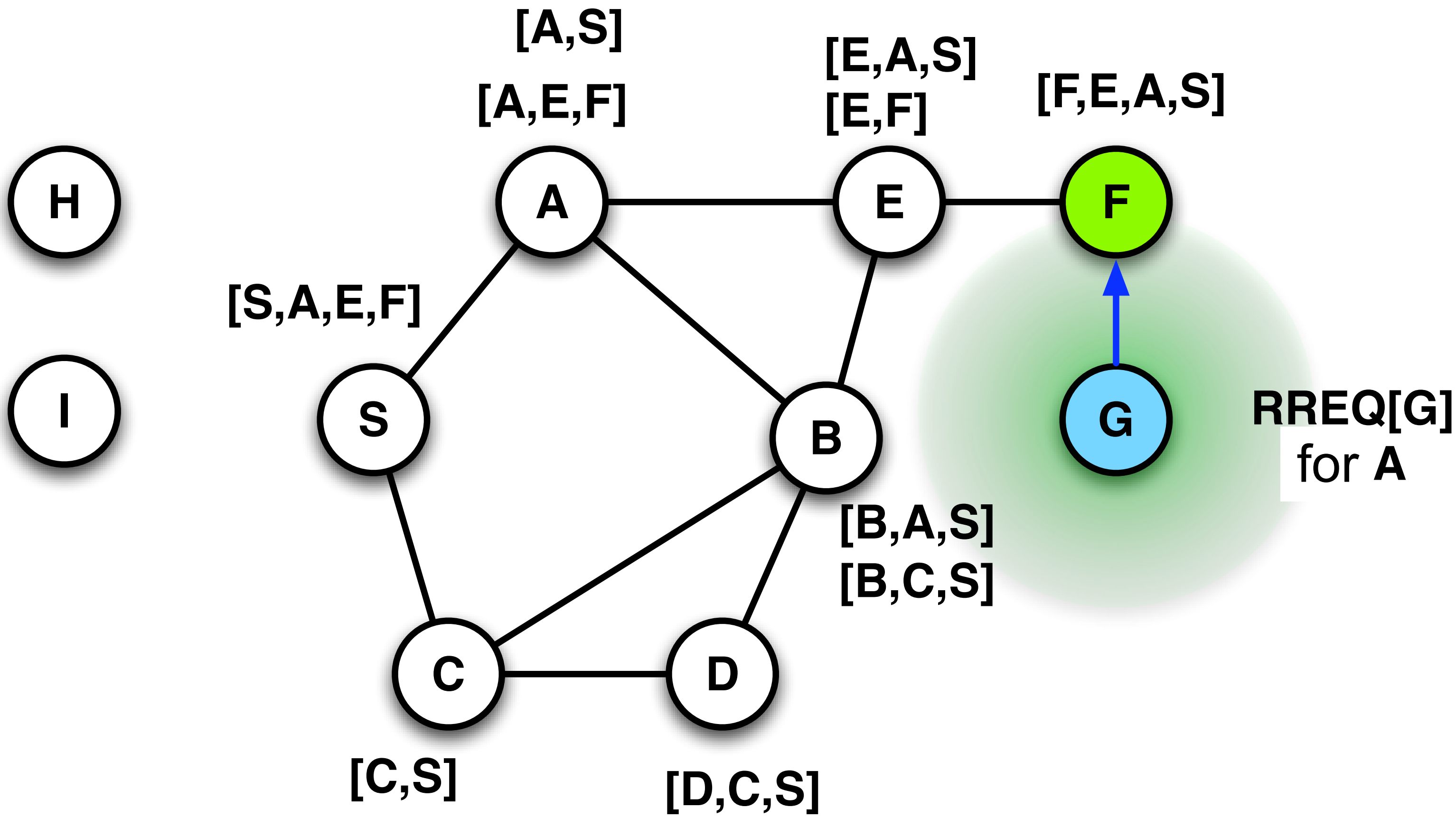
DSR Optimization Route Caching

- **Each node stores information from all available**
  - Header of data packets
  - Route Request
  - Route-Reply
  - partial paths
- **From this information, a route reply is generated**

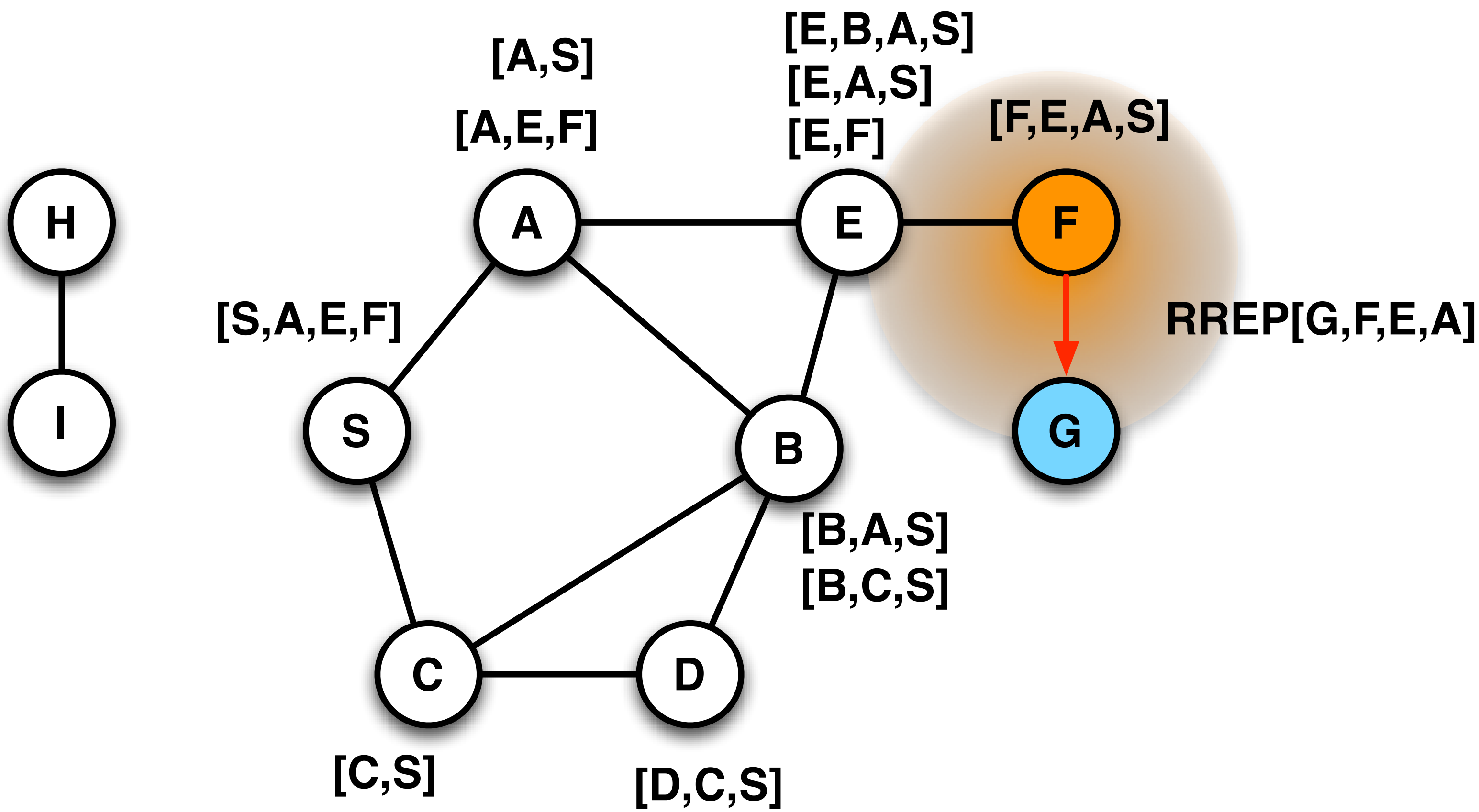
DSR Route Caching



DSR Route Caching

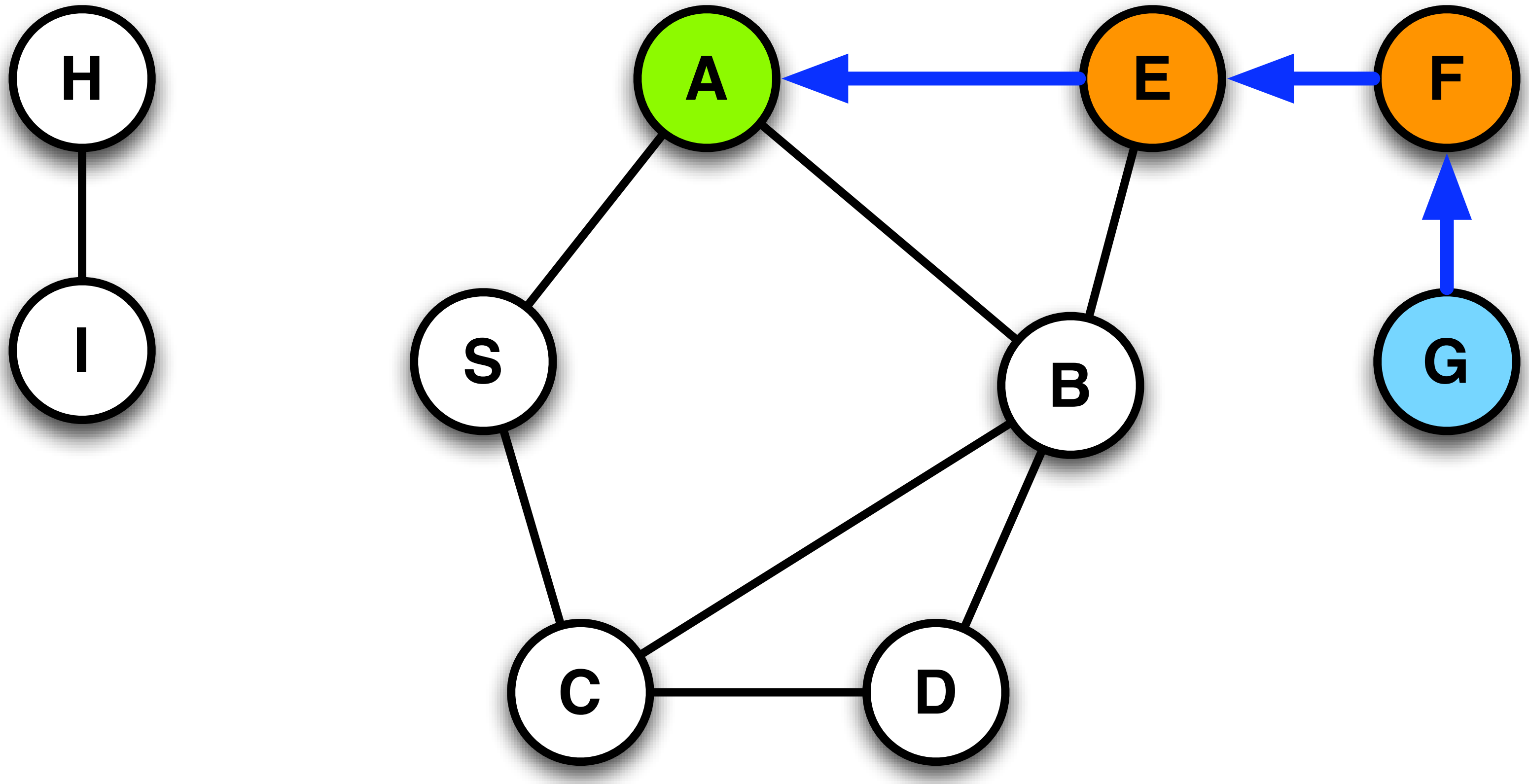


DSR Route Caching



DSR Route Caching

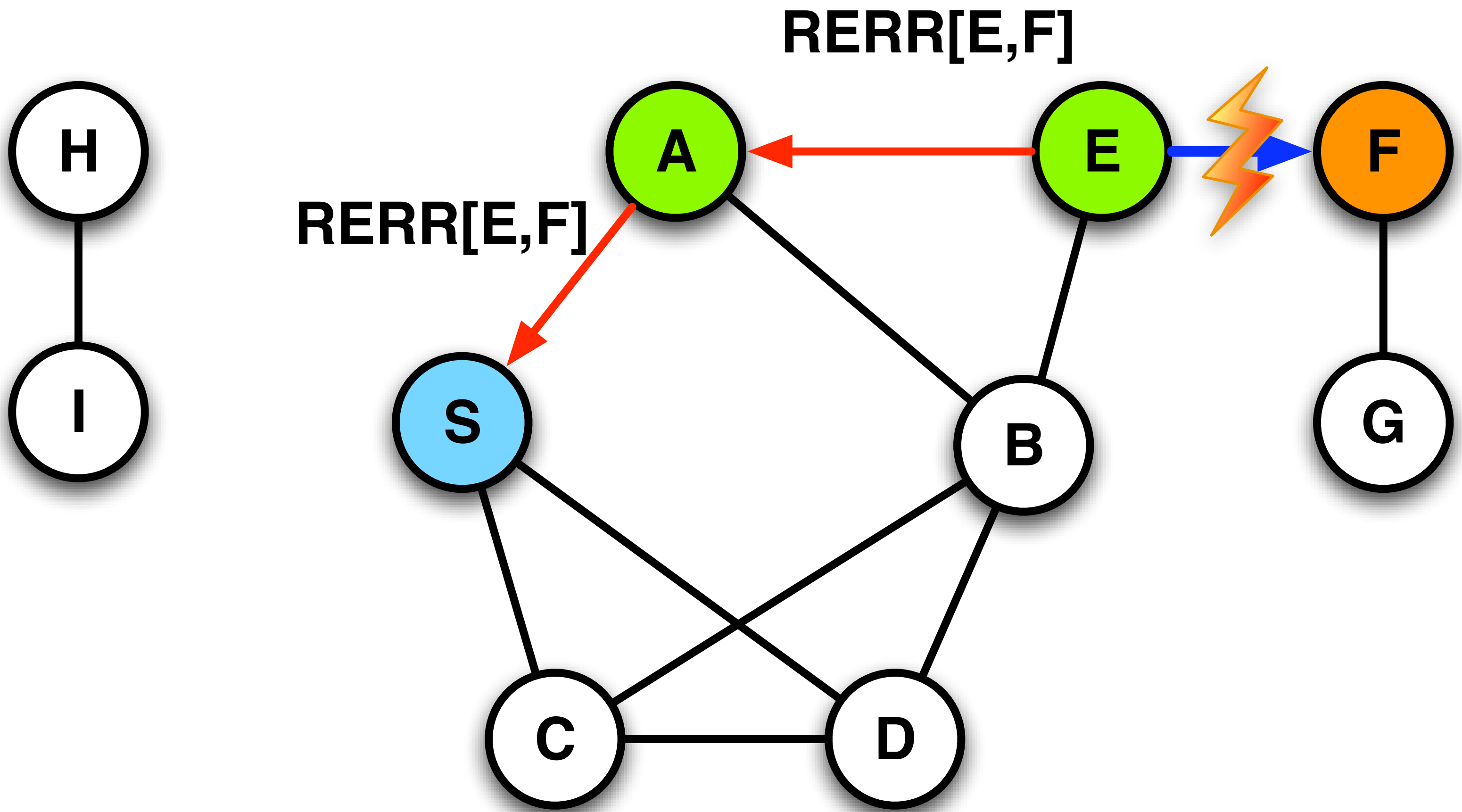
Data packet [G,F,E,A]



DSR Optimization
Route Caching

- **If any information is incorrect**
  - because a route no longer exists
  - then this path is deleted from the cache
  - alternative paths are used
  - or RREQ is generated
- **Missing links are distributed by (RERR) packets in the network**

Route Error



DSR Discussion

- **Benefits**
  - Routes are maintained only between communicating nodes
  - Route caching reduces route search
  - Caches help many alternative routes to find
- **Disadvantages**
  - Header size grows with distance
  - Network may be flooded with route requests
  - Route-Reply-Storm
  - Outdated information may cause cache overhead

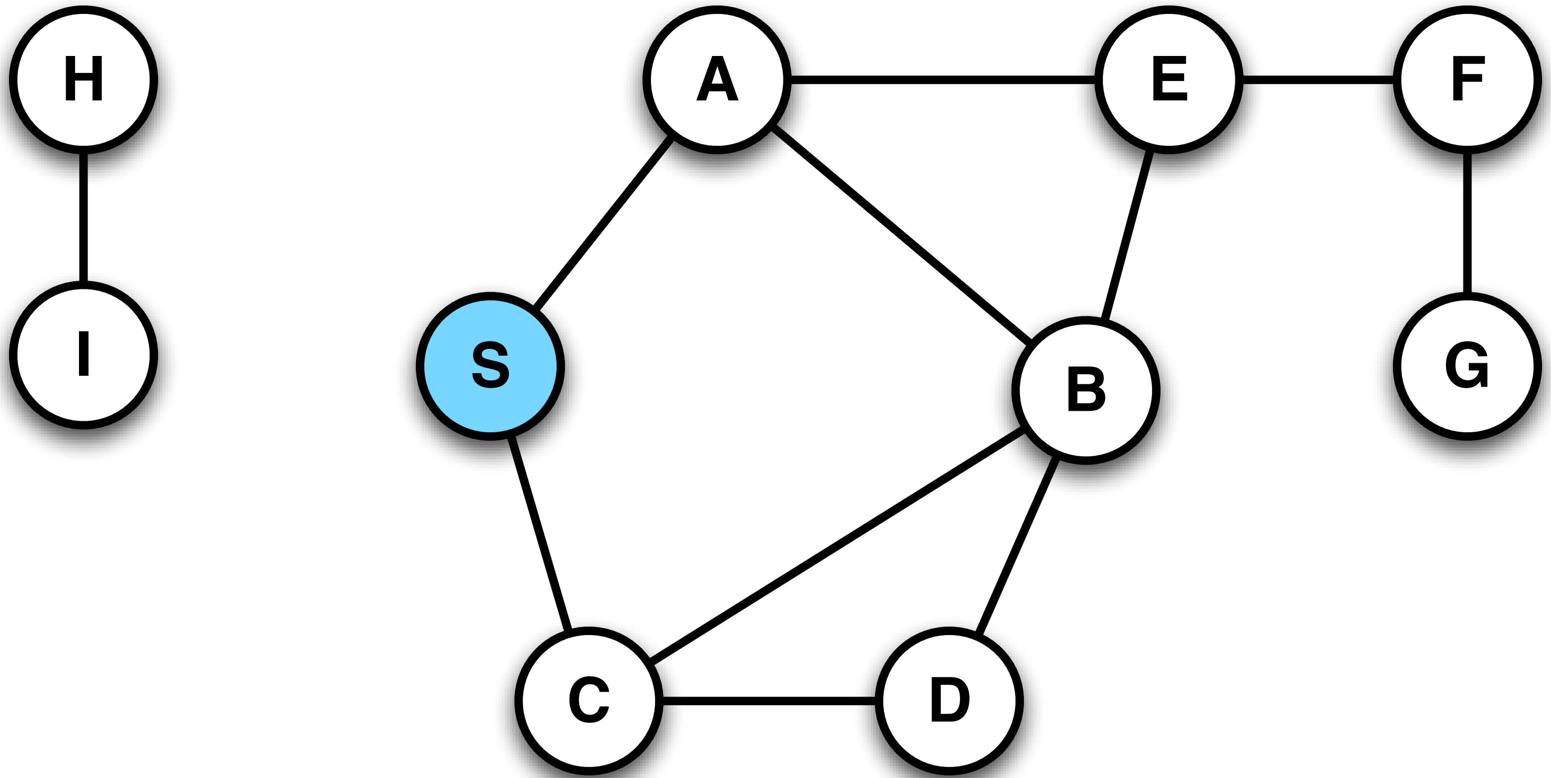
AODV

- **Perkins, Royer**
  - Ad hoc On-Demand Distance Vector Routing, IEEE Workshop on Mobile Computing Systems and Applications,1999
- ~~**Reaktives Routing-Protokoll**~~
- **Reactive routing protocol**
  - Improvement of DSR D
  - no source routing
  - Distance Vector Tables
    - but only for nodes with demand
  - Sequence number to help identify outdated cache info
  - Nodes know the origin of a packet and update the routing table

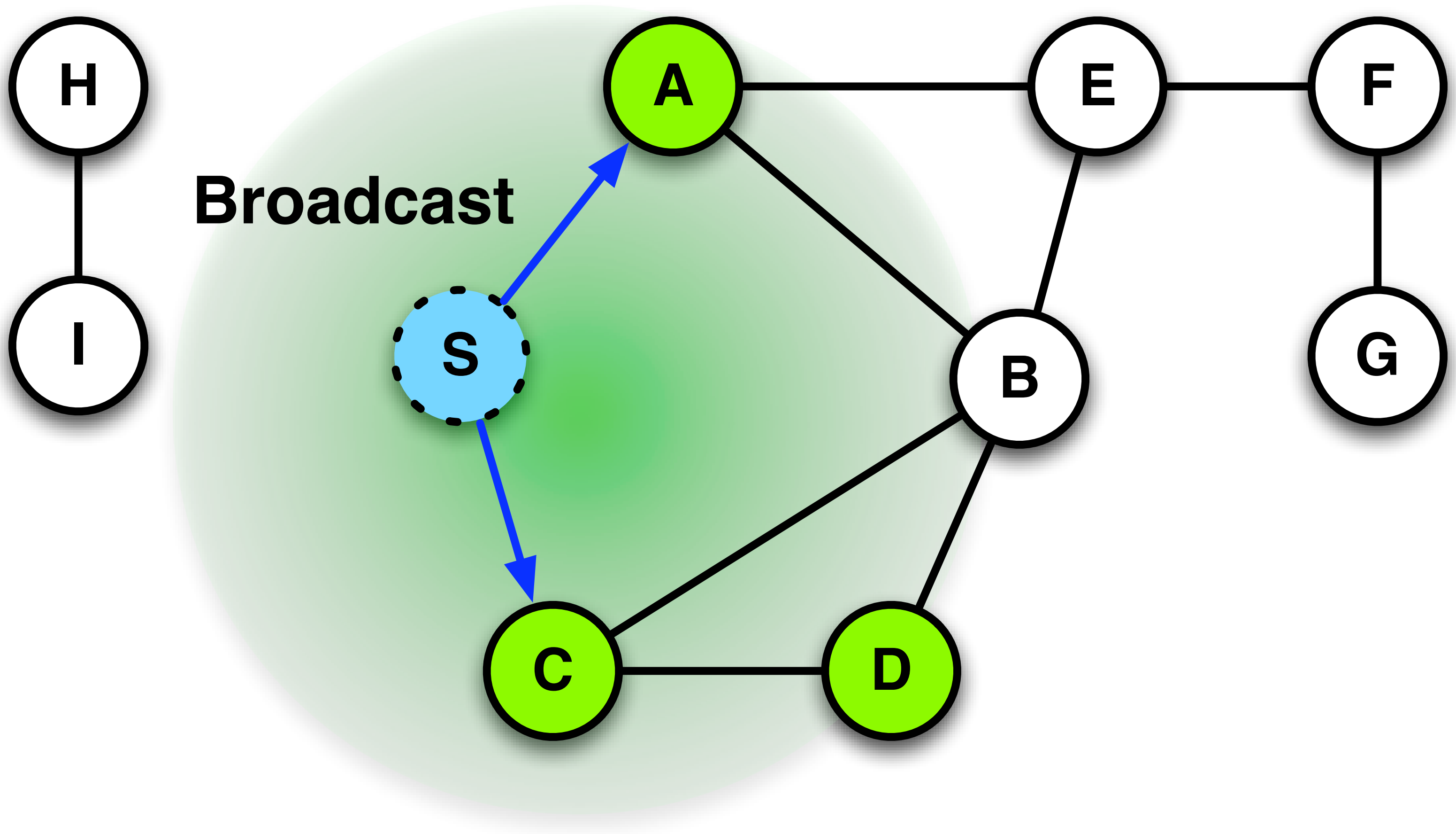
AODV

- **Algorithm**
  - Route Request (RREQ) like in DSR
  - Intermediate nodes set a reverse pointer towards thesender
  - If the target is reached, a Route Reply (RREP) is sent
  - Route Reply follow the pointers
- **Assumption: symmetric connections**

AODV: Example

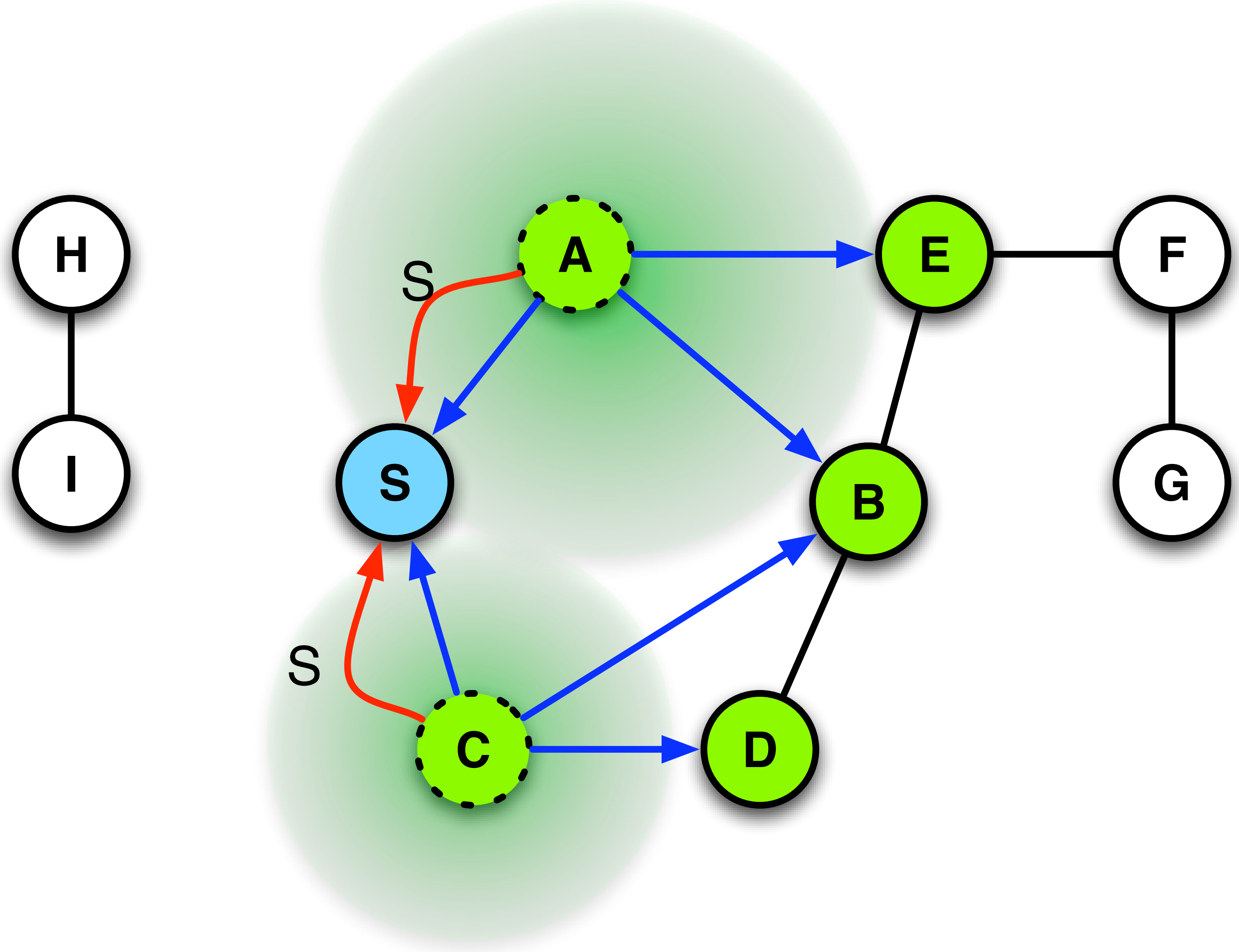


AODV: Example

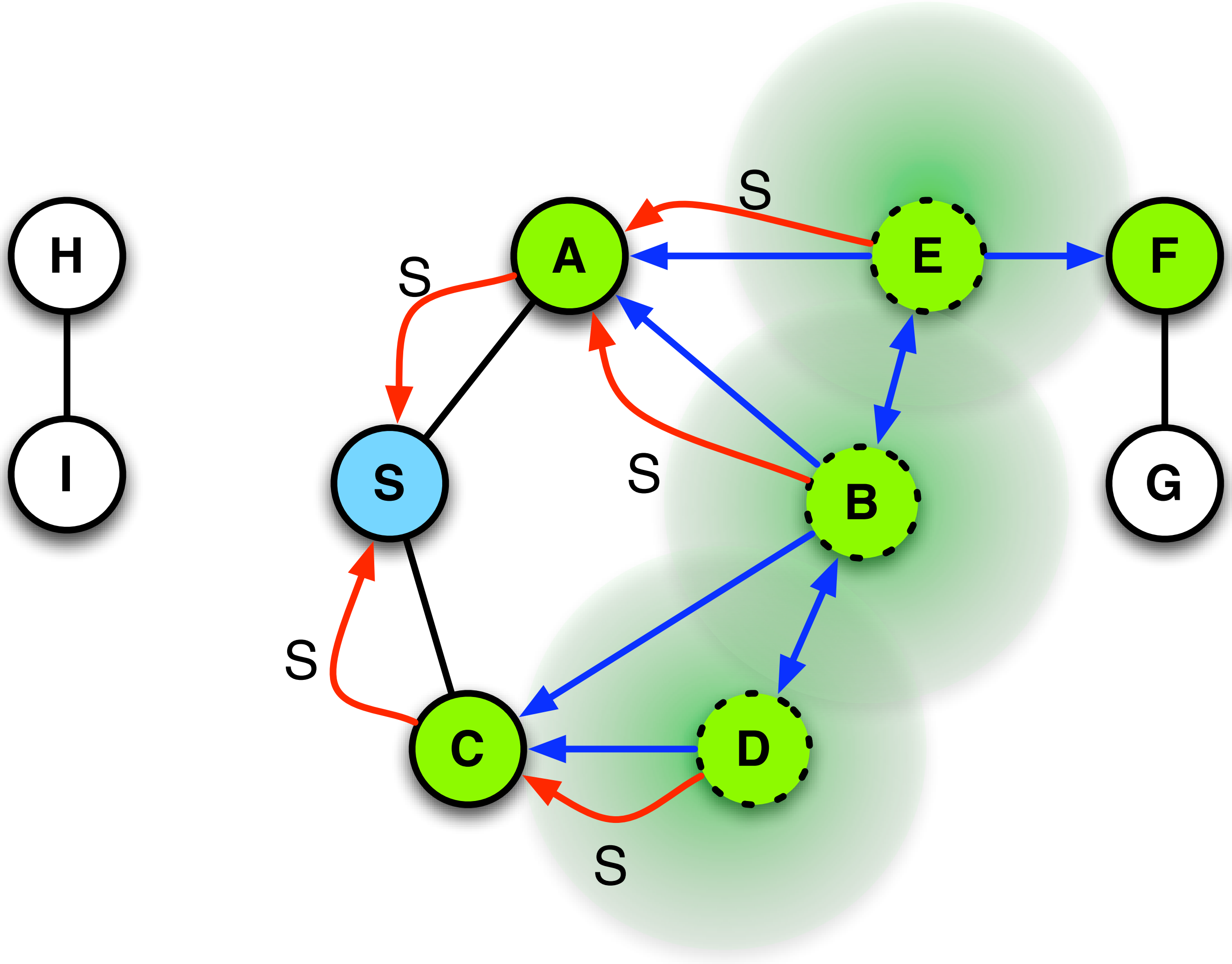


AODV: Example

A	via
	A
S	A
?	–
?	–

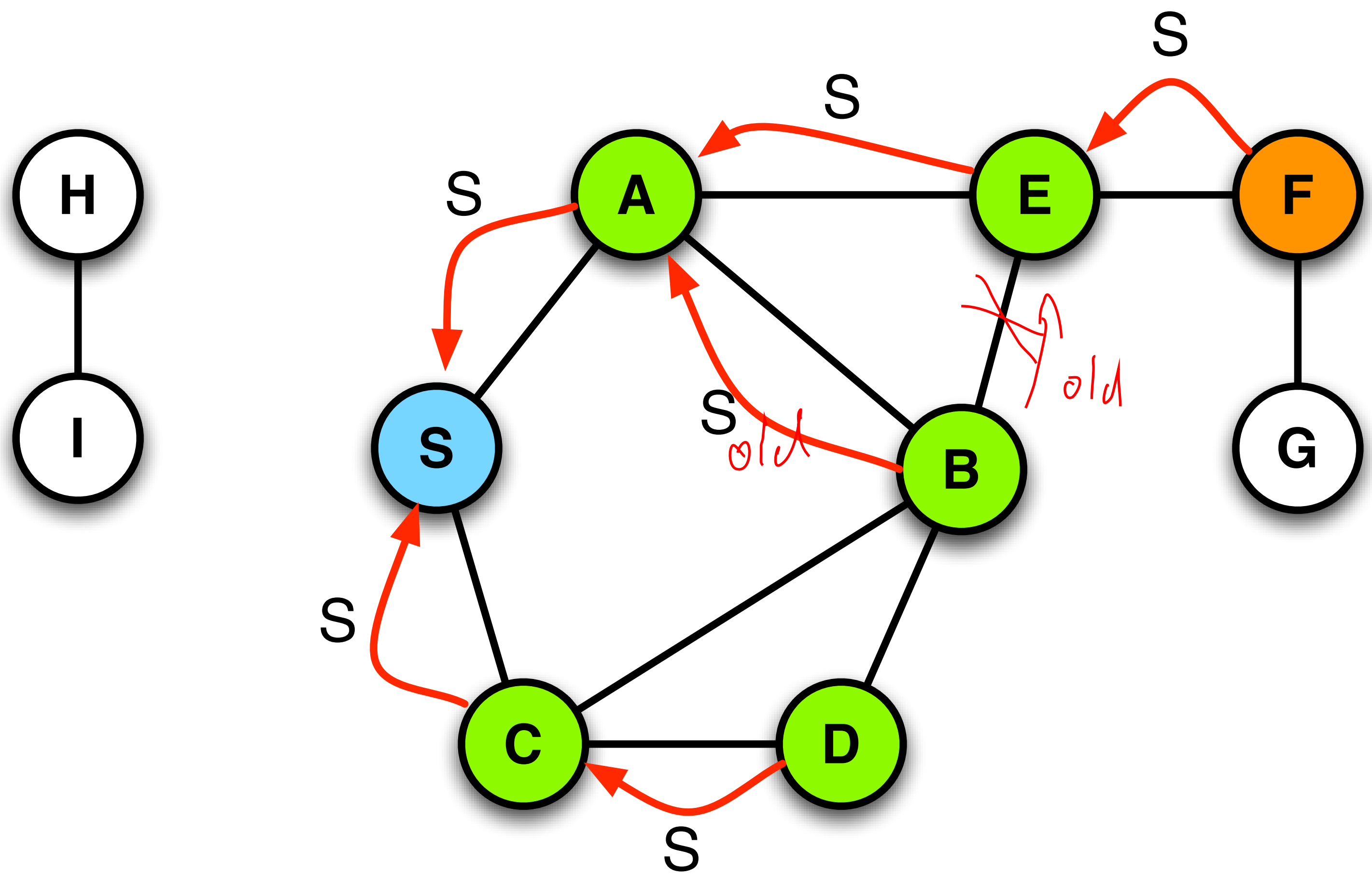


AODV: Example

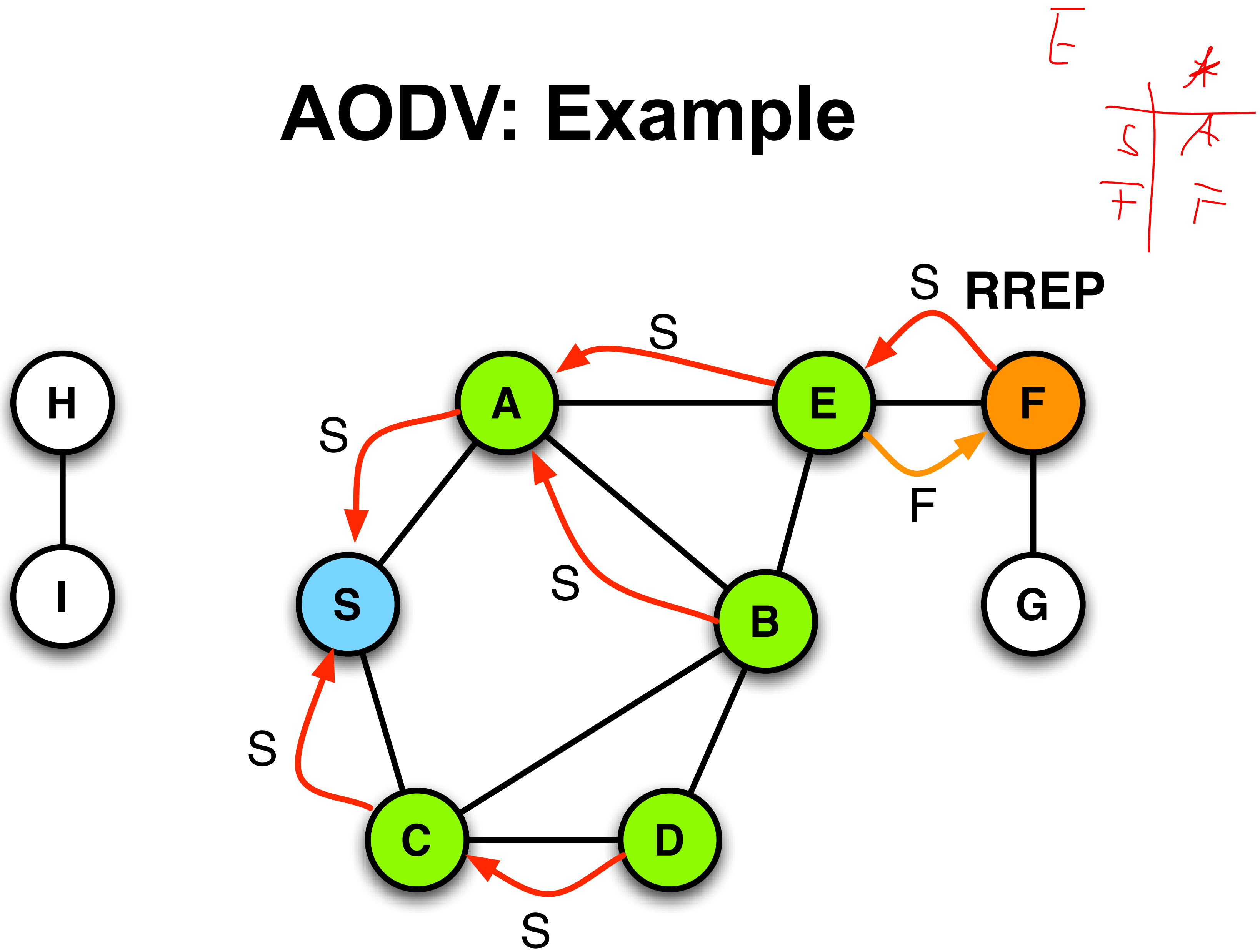


AODV: Example

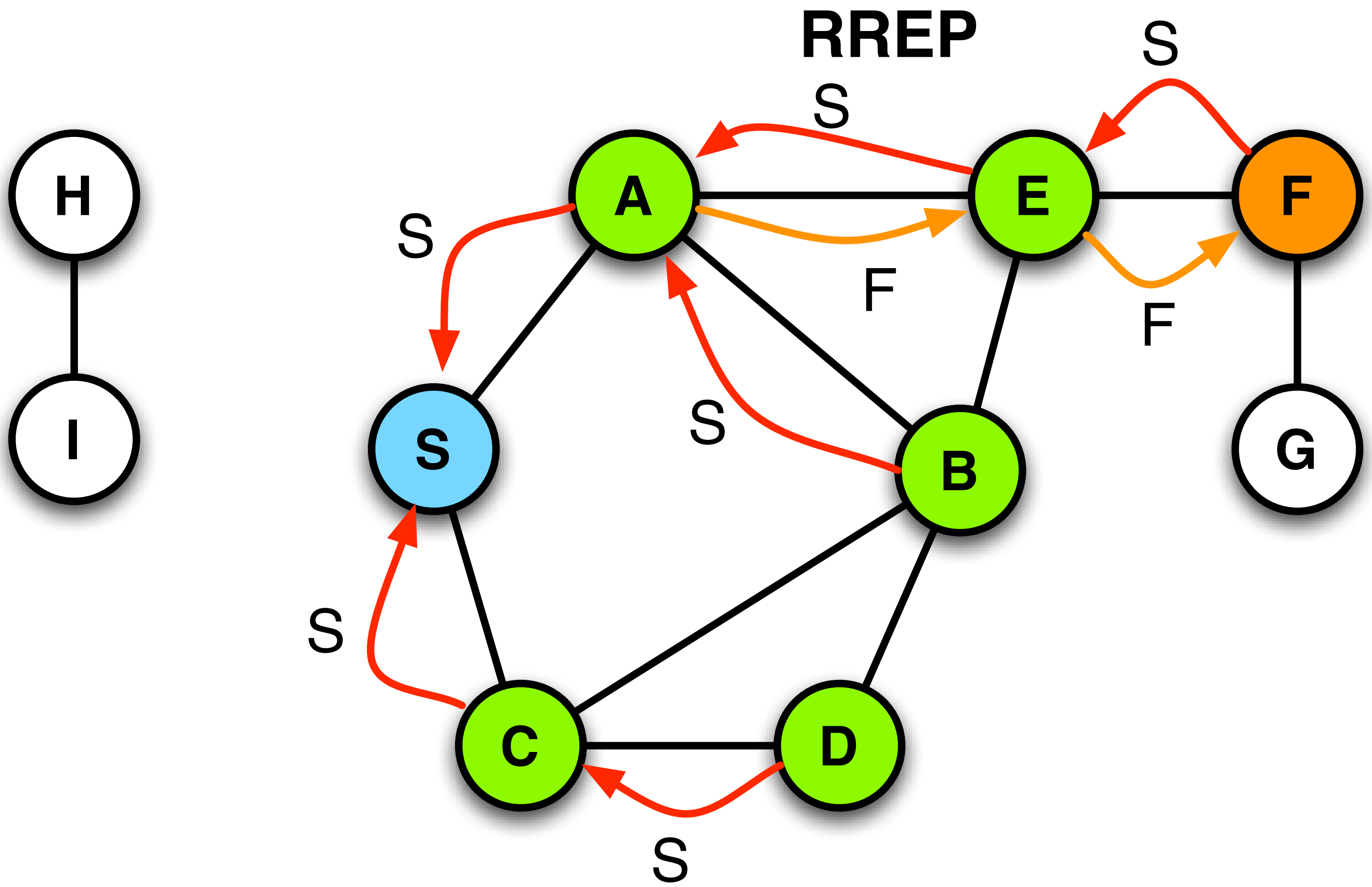
F | via
S | E



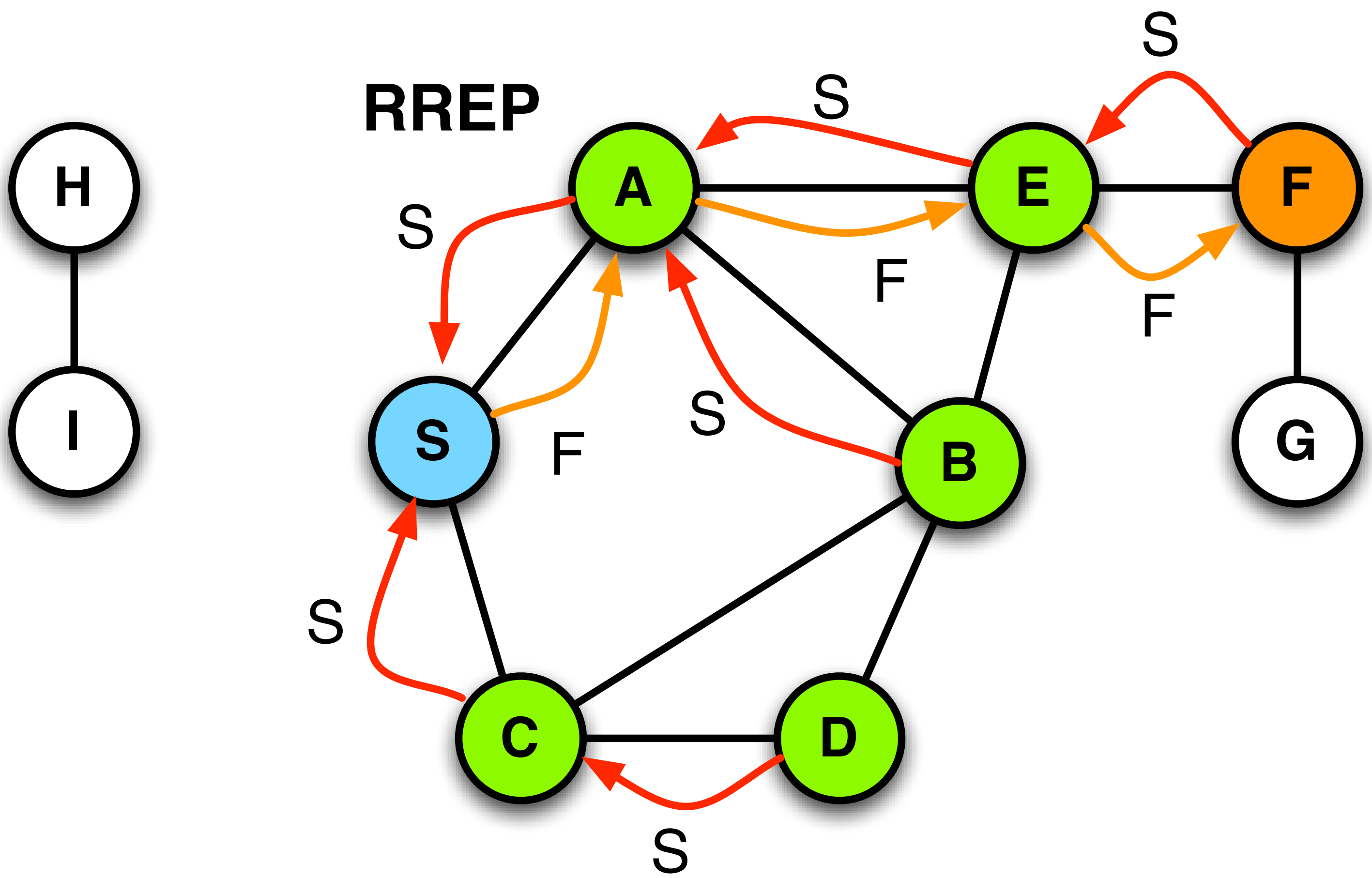
AODV: Example



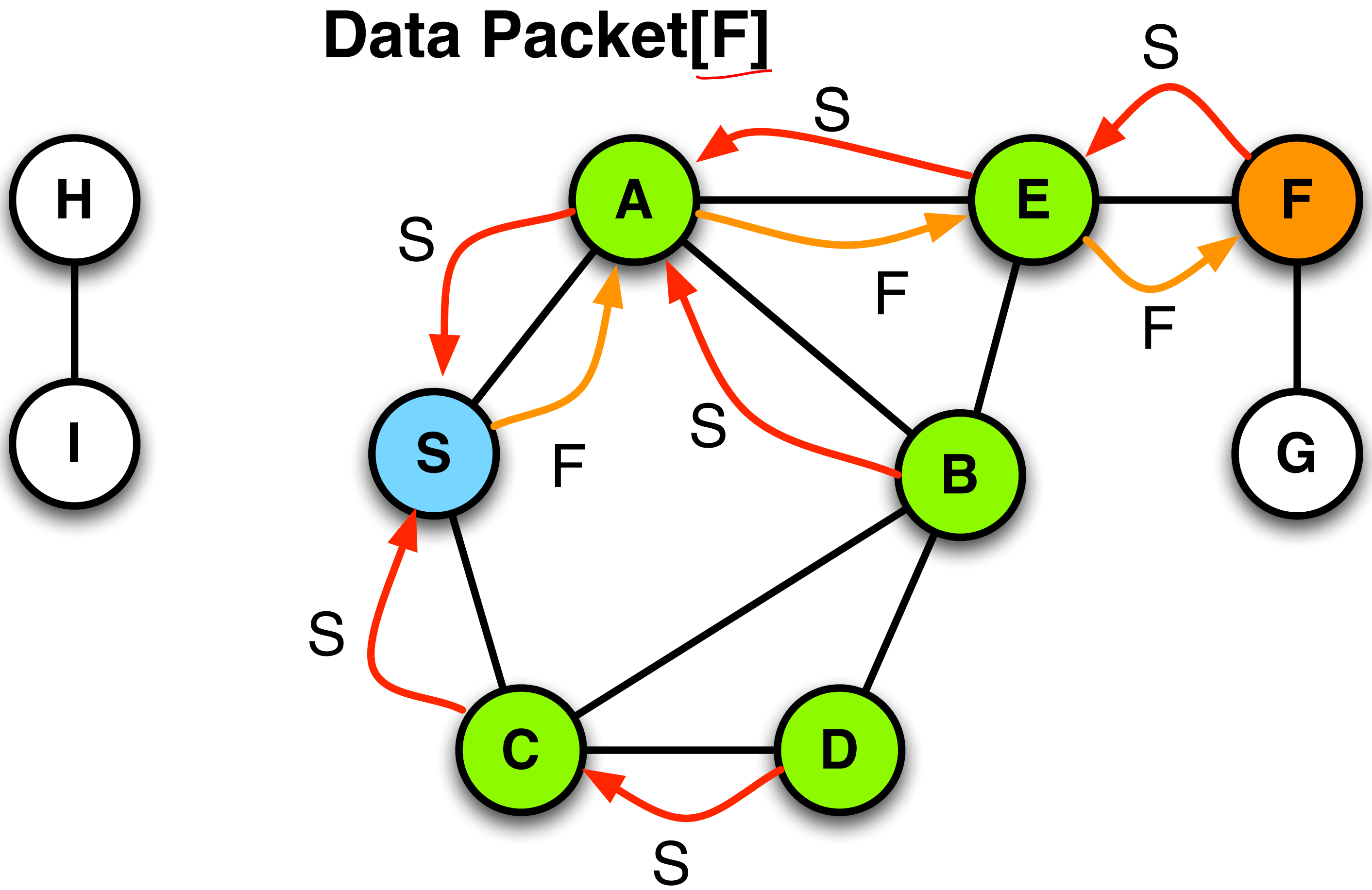
AODV: Example



AODV: Example



AODV: Example



Route Reply in AODV

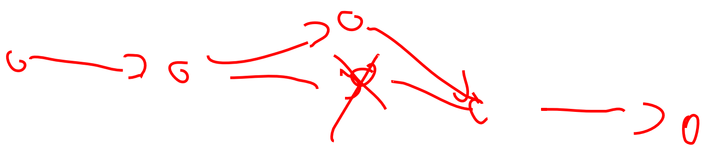
- **Intermediate nodes**
  - may send route-reply packets, if their cache information is up-to-date
- **Destination Sequence Numbers**
  - measure the up-to-dateness of the route information
  - AODV uses cached information less frequently than DSR
  - A new route request generates a greater destination sequence number
  - Intermediate nodes with a smaller sequence number may not generate a route reply (RREP) packets

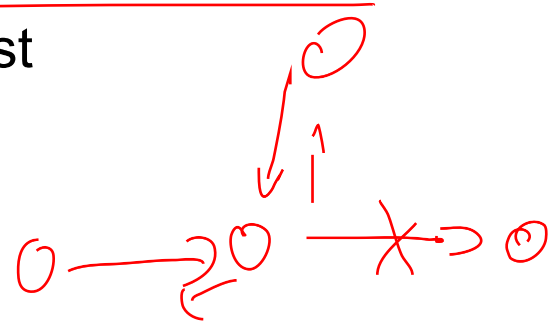
Timeouts

- **Reverse pointers are deleted after a certain time**
  - RREP timeout allows the transmitter to go back
- **Routing table information to be deleted**
  - if they have not been used for some time
  - Then a new RREQ is triggered

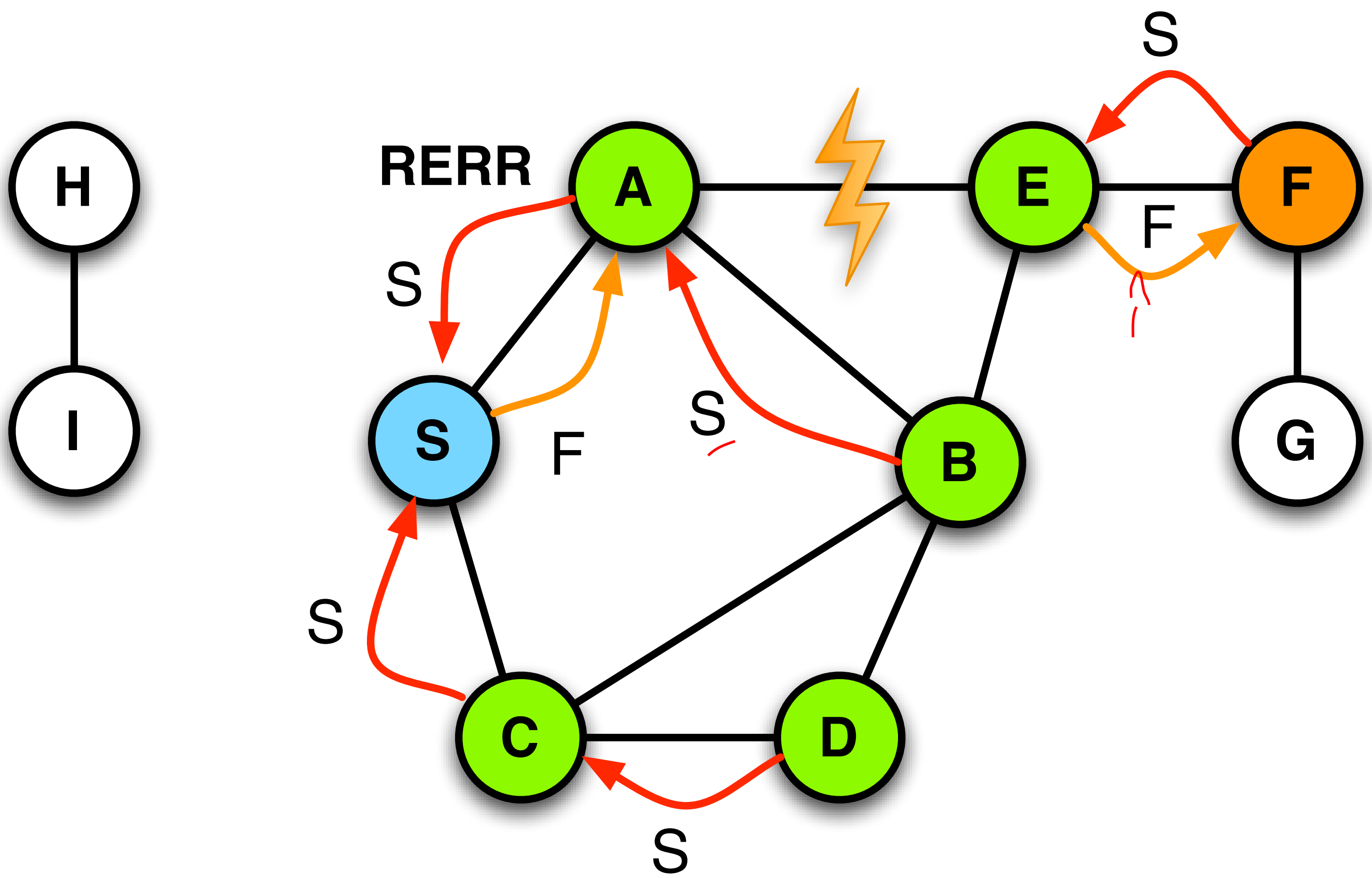
Link Failure Reporting

- **Neighbors of a node X are active,**
  - if the routing table cache are not deleted
- **If a link of the routing table is interrupted,**
  - then all active neighbors are informed
- **Link failures are distributed by Route Error (RERR) packets to the sender**
  - also update the Destination Sequence Numbers
  - This creates new route request





AODV: Example



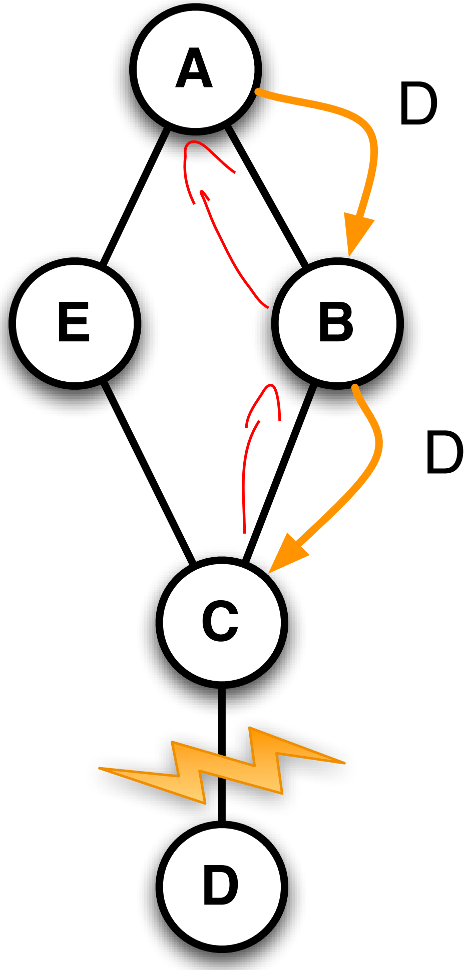
Detection of Link Failure

$\longleftrightarrow$ proactive

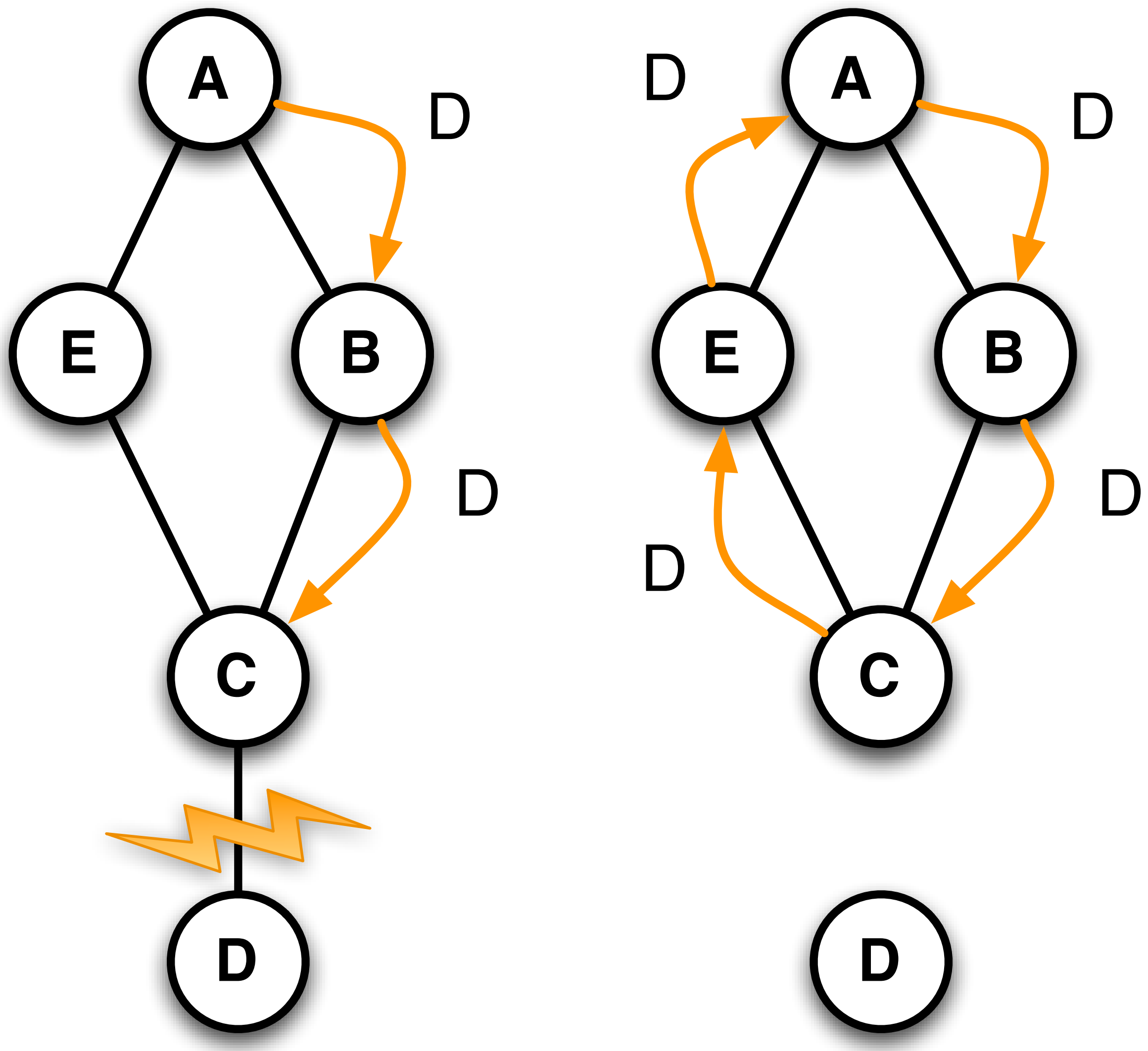
- **Hello messages**
  - neighboring nodes periodically exchange hello packets from
  - Absence of this message indicates link failure
- **Alternative**
  - use information from MAC protocol

Sequence Numbers

- **When a node receives a message with destination sequence number N**
  - then this node sets its number to N
  - if it was smaller before
- **In order to prevent loops**
  - If A has not noticed the loss of link (C, D)
    - (for example, RERR is lost)
  - If C sends a RREQ
    - on path C-E-A
  - Without sequence numbers, a loop will be constructed
    - since A "knows" a path to D, this results in a loop (for instance, CEABC)



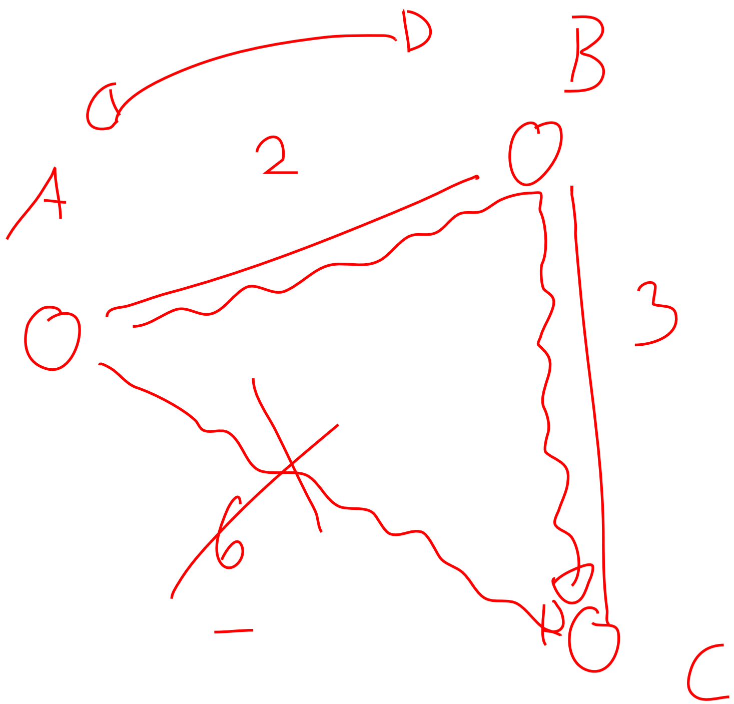
Sequence Numbers

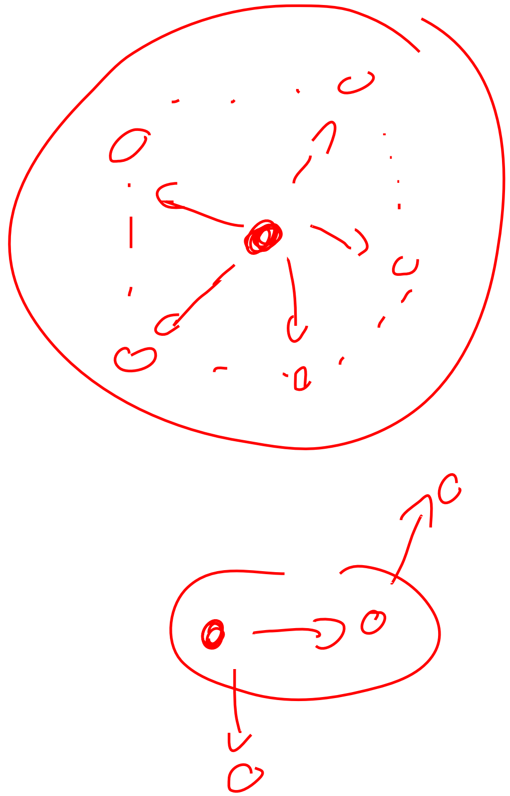


Optimization
Expanding Ring Search

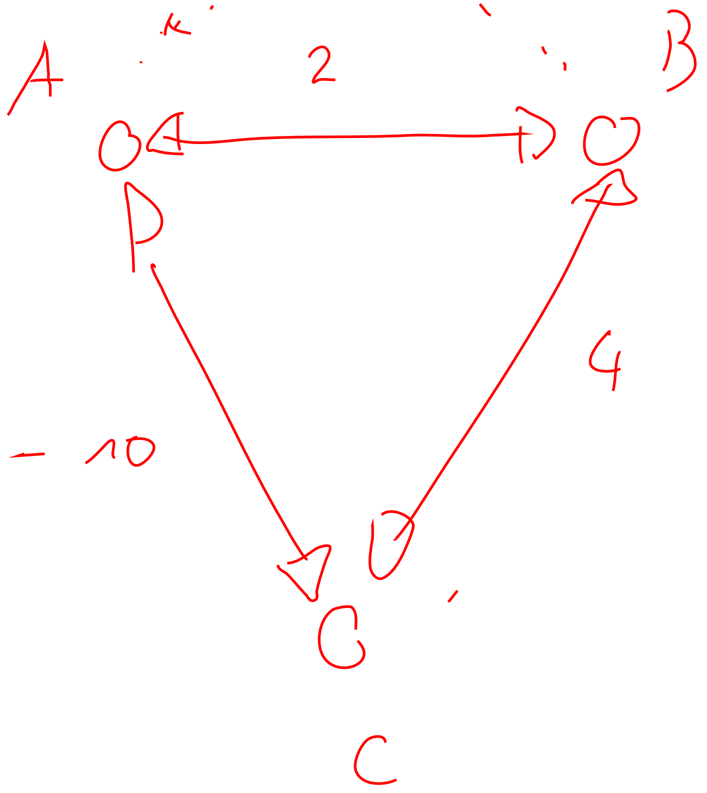
- **Route Requests**
  - *start with small time-to-live value (TTL)*
  - if no Route Reply (RREP) is received, the value is increased by a constant factor and resent
- **This optimization is also applicable for DSR**

Relax





Bellman Ford



A → C -10

A → B 2

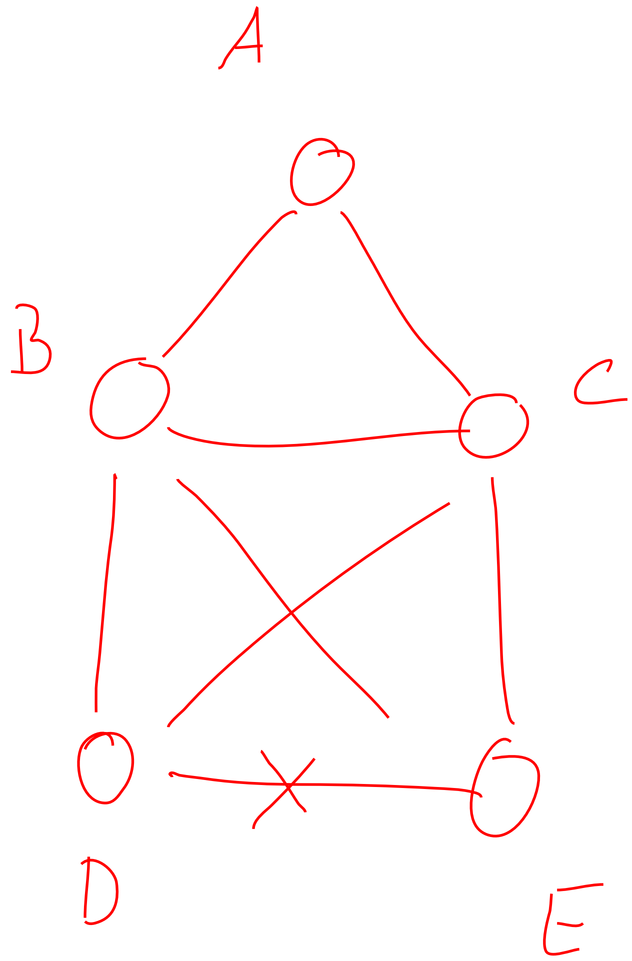
B → A : ~~A~~ -8

B → C : ~~X~~ -6

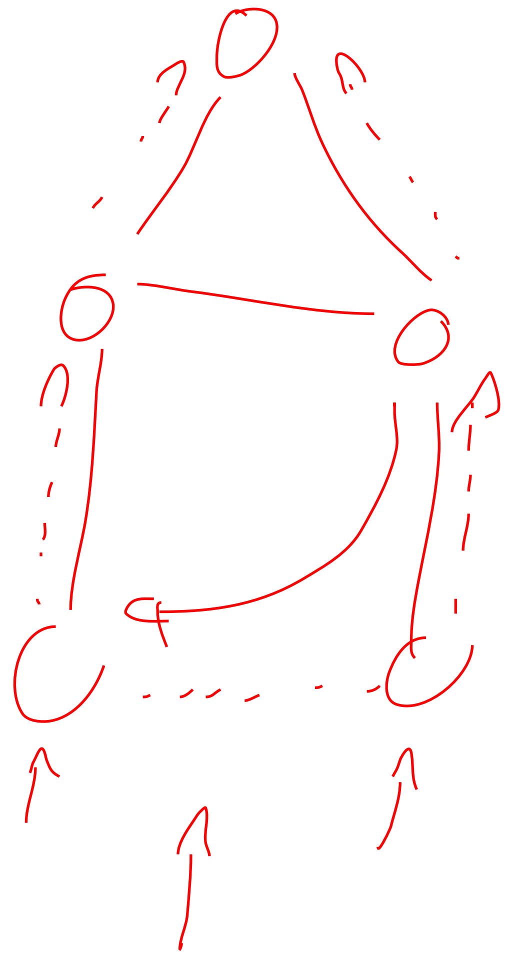
C → A : -10

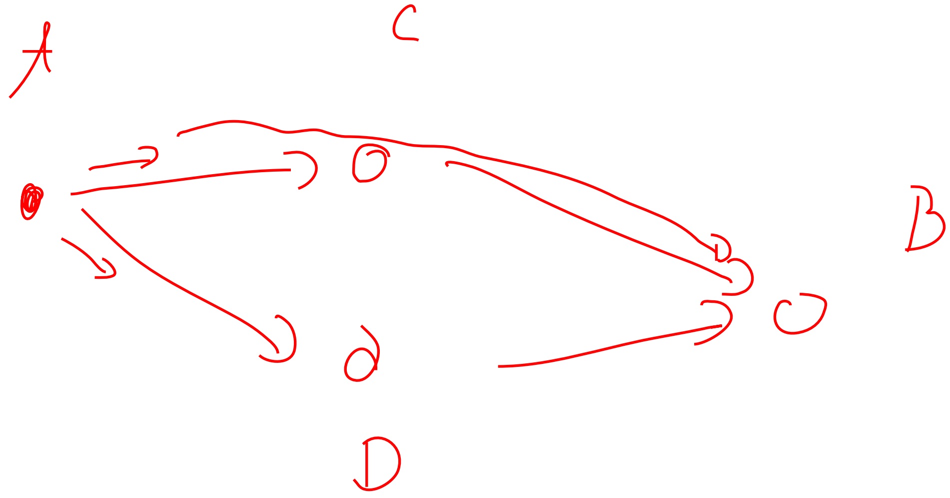
C → B : ~~4~~ -4

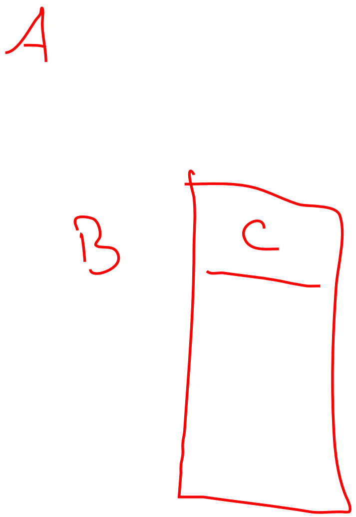
negative cost cycle



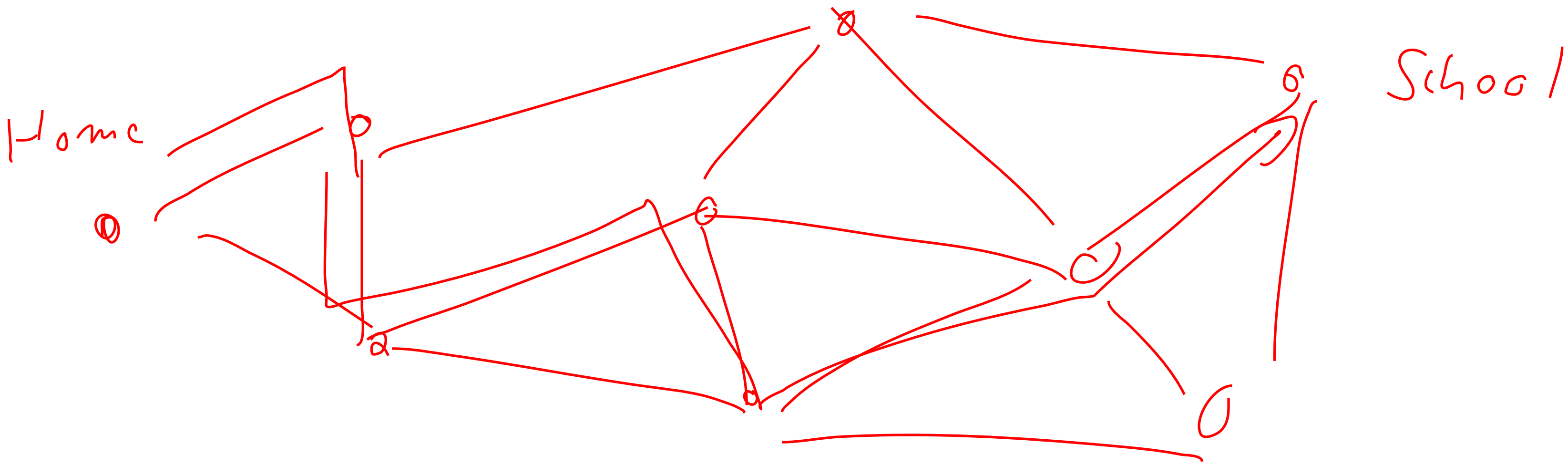
A
B
C
D
E

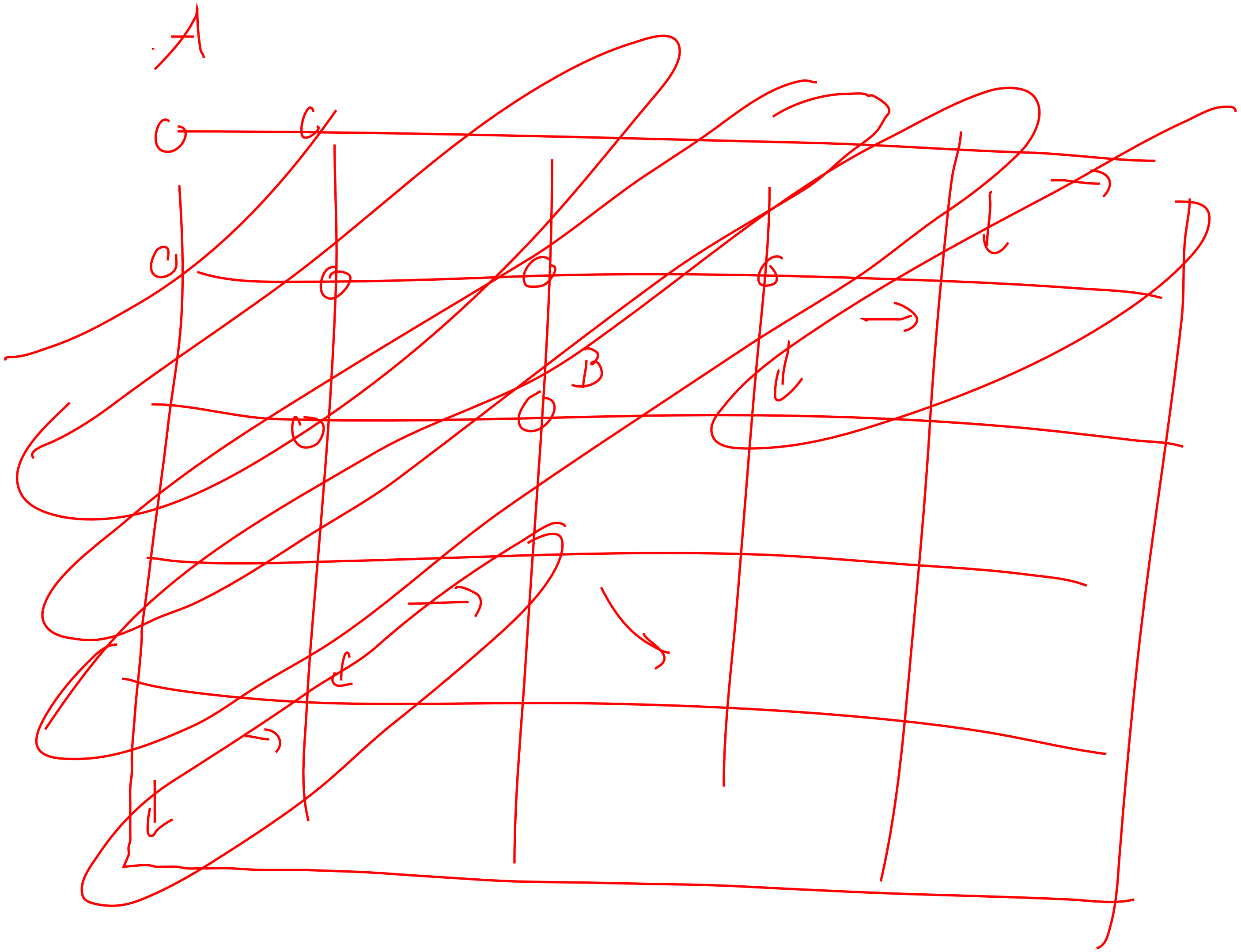


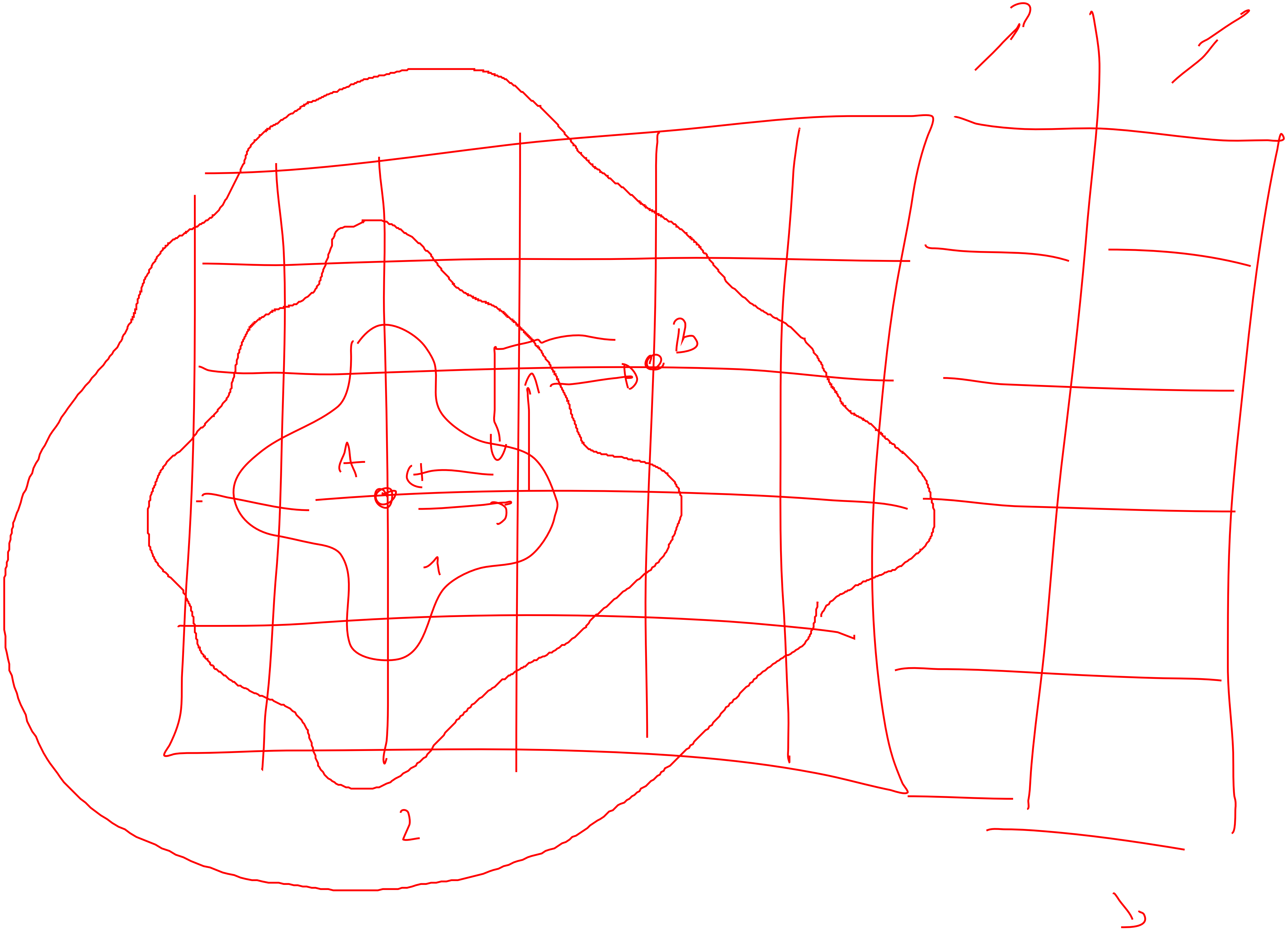

A
C
B
D

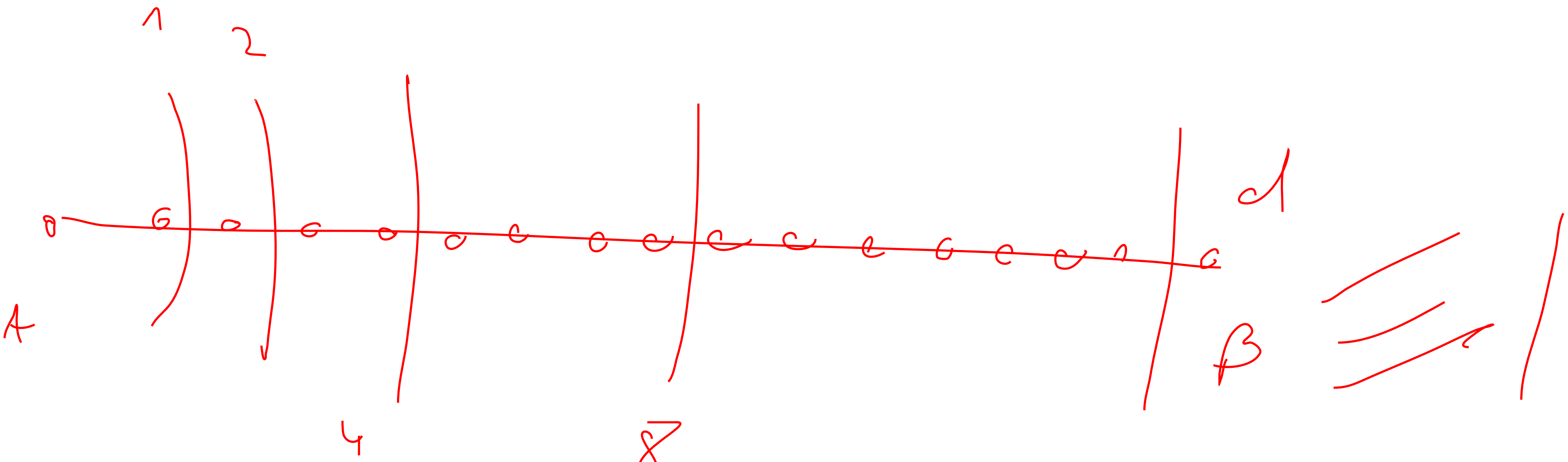

A
B
C

SOURCE ROUTING




A
B





$1+1$

$2+2$

$4+4$

$8+8$

$16+16$

$32+32$

$\lceil \log d \rceil$

$$\sum_{i=1}^{i} 2^i \cdot 2 = 2 \cdot 2^{\log d + 1}$$

$$= 2 \cdot 2 \cdot 2^{\log d}$$

$$= 4 \cdot d$$